



Sistemas Informáticos

2009/2010

---

# **Hacia una Nueva Generación de Campus Virtuales: Integración de Plataformas en el Campus Virtual**

---

Ambrín Chaudhary Tabassam

Jaime Guimerá Coll

Gabriela Ruiz Escobar

Profesor director:

Antonio Navarro Martín





**Facultad de Informática**

**Universidad Complutense de Madrid**

## **TITULO**

Hacia una Nueva Generación de Campus Virtuales: Integración de Plataformas en el Campus Virtual

## **AUTORES**

Chaudhary Tabassam, Ambrín

Guimerá Coll, Jaime

Ruiz Escobar, Gabriela

## **PROFESOR DIRECTOR**

Navarro Martín, Antonio

## **CURSO ACADEMICO**

2009/2010

## **ASIGNATURA**

Sistemas Informáticos



# Resumen

Esta memoria resume el trabajo realizado durante el desarrollo del proyecto “Hacia una Nueva Generación de Campus Virtuales: Integración de Plataformas en el Campus Virtual”, en el contexto de la asignatura Sistemas Informáticos de la Universidad Complutense de Madrid.

El impacto de las tecnologías de la información y la comunicación (TICs) en la educación superior ha generalizado el uso de campus virtuales en las instituciones educativas. Los campus virtuales son complejas aplicaciones web utilizadas por dichas instituciones para facilitar sus relaciones educativas, de investigación y de gestión con y entre estudiantes y profesores. Así, la gestión y mantenimiento de estos campus virtuales conlleva tener en cuenta distintos aspectos nada triviales: soporte a distintas plataformas de e-learning, integración con el resto de servicios de la universidad, diseño de una infraestructura de sistemas hardware capaz de soportar el uso concurrente por una gran cantidad de usuarios, uso de mecanismos eficientes de almacenamiento y recuperación de información, diseño de un formato de datos que facilite la creación y distribución de objetos de aprendizaje, soporte a una gestión del conocimiento educativo, soporte a una estrategia pedagógica, definición y evaluación de la calidad en campus virtuales, etc. Éstas y otras características hacen necesario un esfuerzo por parte de la comunidad científica para responder a los retos que se derivan del diseño, construcción, despliegue, uso y mantenimiento de los campus virtuales.

En el contexto del proyecto de investigación Arquitecturas Avanzadas en Campus Virtuales (TIN2009-14317-C03-01) este proyecto de Sistemas Informáticos desarrolla una arquitectura de integración basada en servicios web para plataformas de campus virtuales.

El objetivo es desarrollar campus virtuales independientes de la plataforma de e-learning subyacente, ayudando así a eliminar la dependencia actual de los campus virtuales de plataformas concretas.

# Overview

This report summarizes the work done during the development of the project “Towards a New Generation of Virtual Campus: Integration Platforms in the Virtual Campus,” in the context of the course Computer Science Systems at the Universidad Complutense de Madrid.

The impact of Information and Communication Technologies (ICT) in higher education has generalized the use of virtual campuses in the educational institutions. Virtual campuses are complex web applications used by these institutions to improve educational, research and management relations between students, teachers and the institution itself. Thus, management and maintenance of these virtual campuses entails having in mind non-trivial aspects like: support of various e-learning platforms, integration with other university services, design of hardware systems infrastructure capable of supporting the concurrently use by a large numbers of users, use of efficient mechanisms for information storage and retrieval, design of a data format that facilitates the creation and distribution of learning objects, support for educational knowledge management, supporting a pedagogical strategy, definition and evaluation of quality in virtual campuses, etc. These and other features make a necessary effort by the scientific community to response to the challenges arising from the design, construction, deployments, use and maintenance of virtual campuses.

In the context of the research project Advanced Architectures in Virtual Campuses (TIN2009-14317-C03-01), this project of Computer Science Systems develops an integration architect based on web services for virtual campuses platforms.

The aim is to develop a virtual campus independent of any e-learning platform behind, helping to eliminate the current reliance on virtual campus of concrete platforms.

# Índice

1. Introducción .....	11
2. Principales plataformas de soporte al e-learning .....	13
2.1    Introducción .....	13
2.2    Blackboard 9.1 .....	16
2.3    Moodle 2.0 .....	18
2.4    Sakai 2.6 .....	20
2.5    Comparación entre plataformas .....	23
3. Arquitecturas Orientadas a Servicios .....	29
3.1    Introducción .....	29
3.2    Principales estándares SOA .....	30
3.2.1    XML .....	30
3.2.2    SOAP .....	30
3.2.3    WSDL .....	31
3.2.4    UDDI .....	32
3.2.5    WS-SECURITY .....	32
3.3    Implementaciones J2EE .....	32
3.4    Implementaciones PHP .....	34
3.5    Servicios web J2EE frente a servicios web PHP .....	35
4. Integración de Plataformas en Campus Virtuales .....	37
4.1    Introducción .....	37
4.2    Arquitecturas precedentes .....	38
4.3    Arquitectura propuesta .....	39
5. Diseño de la solución .....	41
5.1    Introducción .....	41
5.2    Arquitectura .....	41
5.2.1    Diagrama de Despliegue .....	41
5.3    Diseño campus virtual .....	42
5.3.1    Introducción .....	42
5.3.2    Casos de Uso .....	43
5.3.3    Diagrama entidad relación .....	67

5.4	Diseño Servicios web .....	68
5.4.1	Moodle WS .....	68
5.4.2	Sakai WS .....	70
6.	Conclusiones y trabajo futuro .....	79
	Aportaciones a nuestra formación .....	79
	Conclusiones del trabajo .....	80
	Trabajo futuro.....	81
7.	Bibliografía.....	83
	Apéndice A. Especificación de Requisitos Software .....	87
1	Introducción .....	87
1.1	Propósito .....	87
1.2	Ámbito de sistema.....	87
1.3	Definiciones, Acrónimos y Abreviaturas .....	87
1.4	Referencias .....	88
1.5	Visión general del documento.....	88
2.	Descripción General .....	89
2.1.	Perspectiva del Producto .....	89
2.2.	Funciones del Producto .....	89
2.3.	Características de los Usuarios .....	89
2.4.	Restricciones.....	90
2.5.	Suposiciones y Dependencias.....	90
2.6.	Requisitos Futuros .....	90
3.	Requisitos Específicos.....	91
3.1.	Interfaces Externas .....	91
3.2.	Funciones.....	92
3.3.	Requisitos de Rendimiento .....	105
3.4.	Restricciones de Base de Datos lógica .....	105
3.5.	Restricciones de Diseño .....	106
3.6.	Atributos del Sistema .....	106



# Índice de figuras y tablas

FIGURA 2.1 INTERACION DE USUARIOS CON LMS .....	14
FIGURA 2.2 LMS Y SU GESTION DE CONTENIDOS .....	15
FIGURA 2.3 PLATAFORMA EN EL MERCADO ACTUAL .....	16
FIGURA 2.4 ESTADÍSTICAS DE USO DE LAS PLATAFORMAS .....	17
FIGURA 2. 5 UTILIZACION DE MOODLE POR PAISES. ....	18
TABLA 1. TOP 7 PAISES .....	18
FIGURA 2.6 PLATAFORMA SAKAI Y SUS USUARIOS .....	20
FIGURA 2.7 USO DE SAKAI EN EL MUNDO .....	22
TABLA 2. COMPARATIVA DE PLATAFORMAS .....	28
FIGURA 4.1 ARQUITECTURA ANTERIOR DEL CAMPUS VIRTUAL DE LA UCM.....	38
FIGURA 4.2 ARQUITECTURA ACTUAL DEL CAMPUS VIRTUAL DE LA UCM .....	39
FIGURA 4.3 ARQUITECTURA PROPUESTA PARA LA NUEVA GENERACIÓN DE CAMPUS VIRTUALES .....	40
FIGURA 5.1 DIAGRAMA DE DESPLIEGUE .....	42
FIGURA 5.2 HERENCIA DE USUARIOS .....	43
FIGURA 5.3 CASOS DE USO - USUARIOS.....	44
FIGURA 5.4 DIAGRAMA DE ACTIVIDAD - ACCESO .....	45
FIGURA 5.5 DIAGRAMA DE ACTIVIDAD - SALIDA .....	45
FIGURA 5.6 DIAGRAMA DE ACTIVIDAD – MODIFICAR CONTRASEÑA .....	45
FIGURA 5.7 DIAGRAMA DE ACTIVIDAD – MODIFICAR DATOS.....	46
FIGURA 5.8 DIAGRAMA DE ACTIVIDAD – CONSULTA ASIGNATURAS USUARIO .....	46
FIGURA 5.9 DIAGRAMA DE ACTIVIDAD – NUEVO USUARIO .....	47
FIGURA 5.10 DIAGRAMA DE ACTIVIDAD – ELIMINAR USUARIO.....	48
FIGURA 5.11 CASOS DE USO - ASIGNATURAS .....	49
FIGURA 5.12 DIAGRAMA DE ACTIVIDAD – NUEVA ASIGNATURA .....	49
FIGURA 5.13 DIAGRAMA DE ACTIVIDAD – ALTA USUARIO ASIGNATURA.....	50
FIGURA 5.14 DIAGRAMA DE ACTIVIDAD – BAJA USUARIO ASIGNATURA.....	50
FIGURA 5.15 CASOS DE USO - RECURSOS .....	51
FIGURA 5.16 DIAGRAMA DE ACTIVIDAD – CONSULTA RECURSOS ASIGNATURA.....	51
FIGURA 5.17 DIAGRAMA DE ACTIVIDAD – AGREGAR RECURSO ASIGNATURA .....	52
FIGURA 5.18 DIAGRAMA DE ACTIVIDAD – ELIMINAR RECURSO ASIGNATURA .....	52
FIGURA 5.19 CASOS DE USO - FOROS.....	53
FIGURA 5.20 DIAGRAMA DE ACTIVIDAD – CONSUTLA TEMAS FORO ASIGNATURA .....	53
FIGURA 5.21 DIAGRAMA DE ACTIVIDAD – CONSULTA MENSAJES TEMA.....	54
FIGURA 5.22 DIAGRAMA DE ACTIVIDAD – AGREGAR TEMA FORO .....	54
FIGURA 5.23 DIAGRAMA DE ACTIVIDAD – AGERGAR MENSAJES TEMA.....	55
FIGURA 5.24 DIAGRAMA DE ACTIVIDAD – ELIMINAR TEMA FORO .....	55
FIGURA 5.25 DIAGRAMA DE ACTIVIDAD – ELIMINAR MENSAJE FORO .....	56
FIGURA 5.26 CASOS DE USO - ANUNCIOS.....	56
FIGURA 5.27 DIAGRAMA DE ACTIVIDAD – CONSULTA ANUNCIOS ASIGNATURA .....	57
FIGURA 5.28 DIAGRAMA DE ACTIVIDAD – AGREGAR ANUNCIO ASIGNATURA .....	57
FIGURA 5.29 DIAGRAMA DE ACTIVIDAD – ELIMINAR ANUNCIO ASIGNATURA .....	58
FIGURA 5.30 CASOS DE USO - PLATAFORMA .....	58
FIGURA 5.31 DIAGRAMA DE ACTIVIDAD – CONSULTA USUARIOS .....	59
FIGURA 5.32 DIAGRAMA DE ACTIVIDAD – CONSULTA ASIGNATURAS .....	59
FIGURA 5.33 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA PRESENTACIÓN.....	60
FIGURA 5.34 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (ACTIONS 1) .....	61
FIGURA 5.35 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (ACTIONS 2) .....	61
FIGURA 5.36 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (FORMS).....	62
FIGURA 5.37 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (BEANS) .....	63
FIGURA 5.38 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (BBDD) .....	63

FIGURA 5.39 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (UTIL).....	64
FIGURA 5.40 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA INTEGRACION.....	65
FIGURA 5.41 DIAGRAMA DE SECUENCIA. CAMPUS VIRTUAL – ALTA ASIGNATURA .....	66
FIGURA 5.42 DIAGRAMA DE ENTIDAD RELACION .....	67
FIGURA 5.43 DIAGRAMA DE CLASES. MOODLE WS – CAPA LOGICA NEGOCIO .....	69
FIGURA 5.44 DIAGRAMA DE CLASES. MOODLE WS – CAPA INTEGRACION .....	69
FIGURA 5.45 DIAGRAMA DE SECUENCIA. MOODLE WS – ALTA ASIGNATURA .....	70
TABLA 3. FUNCIONALIDADES DESARROLLADAS EN SAKAIWS .....	73
FIGURA 5.46 DIAGRAMA DE CLASES. SAKAI WS – CAPA LOGICA NEGOCIO.....	74
FIGURA 5.47 DIAGRAMA DE CLASES. SAKAI WS – CAPA INTEGRACION .....	75
FIGURA 5.48 DIAGRAMA DE SECUENCIA. SAKAI WS – ALTA ASIGNATURA .....	78
FIGURA 6.1 CONCLUSION.....	80
FIGURA A.1 DIAGRAMA E-R. BASE DE DATOS DEL CAMPUS VIRTUAL.....	106

# 1. INTRODUCCIÓN

El uso de las Tecnologías de Información y Comunicación (TIC) en la educación ha sido un foco de interés para la comunidad educativa durante años (Guri-Rosemblit 05). En el presente, el uso de TICs en la educación se denomina generalmente como aprendizaje electrónico o e-learning (Kaplan-Leison 09). El e-learning se ha convertido en un factor clave para el éxito de universidades e instituciones de educación superior (Guri-Rosemblit 05, Neville 05). Por este motivo, las universidades tradicionales están evolucionando, total o parcialmente, hacia universidades virtuales, que ofrecen sus cursos en un formato e-learning (PLS 04, Whittington 98).

En este contexto aparecen los campus virtuales como medio para ofrecer el e-learning tanto a alumnos como a profesores, favoreciendo la comunicación y el desarrollo de actividades entre los participantes de un curso. Entre estas actividades destacan: presentación de contenidos como apuntes, transparencias o trabajos, foros de comunicación, encuestas, wikis, gestión de calificaciones, etc.

Esta situación ha promovido la aparición de campus virtuales: “El campus virtual es una metáfora para el entorno de la enseñanza, aprendizaje e investigación electrónica creado por la convergencia de varias tecnologías relativamente nuevas incluyendo, pero sin restricciones, internet, WWW (World Wide Web), comunicación a través de ordenadores, videoconferencia, multimedia, trabajo en grupo, video bajo demanda, publicación de escritorios, sistemas de tutoría inteligentes y realidad virtual” (Van Dusen 97). En estudios más recientes (PLS 04, Allison 08, Epper 04, Green 05) los campus virtuales se entienden, en un sentido amplio, como la integración de las TICs tanto en términos educativos como de organización de entornos universitarios.

En este trabajo, los campus virtuales se consideran como los sistemas software que las grandes instituciones educativas usan para dar soporte a sus actividades de enseñanza y aprendizaje. Al tratarse de una definición amplia, la naturaleza de un campus virtual depende del tipo de sistema software que utilice para dar su soporte. Básicamente, se pueden seleccionar tres tipos principales de sistemas: sistemas de gestión de cursos, sistemas de gestión de aprendizaje y sistemas de gestión de contenidos para el aprendizaje.

*Los sistemas de gestión de contenidos* (Course Management Systems, CMS) son sistemas online diseñados originalmente para dar soporte en el aprendizaje en las aulas en el ámbito académico. Los CMSs permiten a los profesores la capacidad de realizar las siguientes tareas: proporcionar recursos del curso online, hacer el seguimiento del progreso de sus estudiantes (a través de funciones de evaluación, panel de discusiones y otras herramientas de comunicación), incluir una caja de seguridad para los estudiantes y permitir el cálculo de estadísticas del curso. Ejemplos de CMS incluyen productos comerciales como *BlackBoard* y *WebCT* (adquirido por BlackBoard), y sistemas de código abierto como *Moodle* (Carliner 05).

*Los sistemas de gestión de aprendizaje* (Learning Management Systems, LMS) son software que fundamentalmente actúan como un registro electrónico mediante la realización, electrónicamente, de diversas inscripciones y tareas relacionadas. Ejemplos de LMSs incluyen productos comerciales como *NetDimensions EKP*, *Saba* y *SumTotal Systems* (Carliner 05). Sin embargo, en el presente, los términos CMS, LMS y VLE (Virtual Learning Environment, Entorno de Aprendizaje Virtual) se usan de forma equivalente.

Los sistemas de gestión de contenidos para el aprendizaje (Learning Content Management Systems, LCMS) son entornos multiusuario donde los desarrolladores pueden crear, almacenar, reutilizar, gestionar y enviar contenidos desde un repositorio central. Ejemplos de

LCMS incluyen productos de código abierto como *ATutor*, y productos comerciales como *Intelligent Web Teacher* y *SumTotal TotalLCMS*.

Estos CMS, LMS y LCMS pueden usarse independientemente o pueden ser parte de una infraestructura, de mayores dimensiones, de gestión universitaria, la *Gestión de Entornos de Aprendizaje* (Management Learning Environments, MLE) (IMS 09).

En este contexto, la Universidad Complutense de Madrid, como muchas otras universidades, está realizando un gran esfuerzo para introducir modernas técnicas de e-learning en los actuales procesos educativos. En el curso 2003-2004 se creó el Campus Virtual de la UCM (CV UCM) (UCM VC 09). El principal objetivo era ofrecer a alumnos y a profesores un entorno que mejorase la calidad de la enseñanza y la investigación, utilizando las modernas Tecnologías de la Información y la Comunicación (TIC) (Navarro 07). El Campus Virtual de la UCM aporta diversas funcionalidades como son la gestión de los estudiantes matriculados en las asignaturas, la gestión de contenidos de las propias asignaturas y herramientas de comunicación como grupos de trabajo, foros y chats. En el curso académico 2008-2009 el tamaño del CV UCM era de 68.000 estudiantes y 5.300 profesores.

Durante estos años uno de los mayores problemas en el Campus Virtual de la UCM ha sido la dependencia de una plataforma de e-learning subyacente. Esta dependencia obliga a: (i) la negociación de una licencia comercial periódicamente (el Campus Virtual de la UCM tiene más de 65.000 usuarios), o alternatively, el uso de un producto no comercial que no ha sido probado en un campus virtual con decenas de miles de usuarios; y (ii) el rechazo de los usuarios a cambiar de plataforma (incluso de una versión a la siguiente), anclando el campus

Para resolver estos problemas surge el proyecto de investigación Arquitecturas Avanzadas en Campus Virtuales (TIN2009-14317-C03-01) que busca desarrollar una nueva arquitectura software para campus virtuales de nueva generación (Navarro 10). El objetivo del proyecto es ser capaz de cumplir con las necesidades de los usuarios creando un campus virtual *de marca blanca* (Navarro 10) independiente de la plataforma subyacente mediante una arquitectura orientada a servicios y que pueda ser parte de un entorno universitario mayor. Para gestionar las necesidades cambiantes de los usuarios de los campus virtuales, el proyecto propone utilizar la arquitectura dirigida por modelos (OMG 01) para su desarrollo. Finalmente, el proyecto prevé el uso de dispositivos avanzados móviles para extender el uso del campus virtual más allá de un acceso basado en ordenador (Navarro 10).

En este contexto de investigación se ha desarrollado este proyecto de Sistemas Informáticos. El objetivo es desarrollar una arquitectura inicial orientada a servicios que permita el desarrollo de un campus virtual de marca blanca. Así, la plataforma subyacente de e-learning está totalmente oculta al usuario. El objetivo no es cambiar en caliente la plataforma en uso, sino permitir el cambio de plataforma de un curso académico a otro sin que el usuario sea consciente de dicho cambio. Por supuesto, para este fin sería necesario el desarrollo de mecanismos de traducción de cursos y materiales entre plataformas. Este último desarrollo queda totalmente fuera de los objetivos del presente proyecto de Sistemas Informáticos.

## 2. PRINCIPALES PLATAFORMAS DE SOPORTE AL E-LEARNING

### 2.1 INTRODUCCIÓN

La constante capacitación es un requerimiento de las sociedades actuales. Con todas las presiones del día a día, utilizar Internet para la capacitación no es una moda sino una herramienta que facilita esta constante actualización (PHC Consultores 10).

Desde la revolución industrial, la educación a distancia surge para facilitar la especialización del personal. Y así con el avanzar de los tiempos, la tecnología siempre ha presentado diferentes opciones para compartir información, desde material impreso, radioconferencias, teleconferencias, videos e incluso CD-ROMs apoyados por correspondencia postal y llamadas telefónicas.

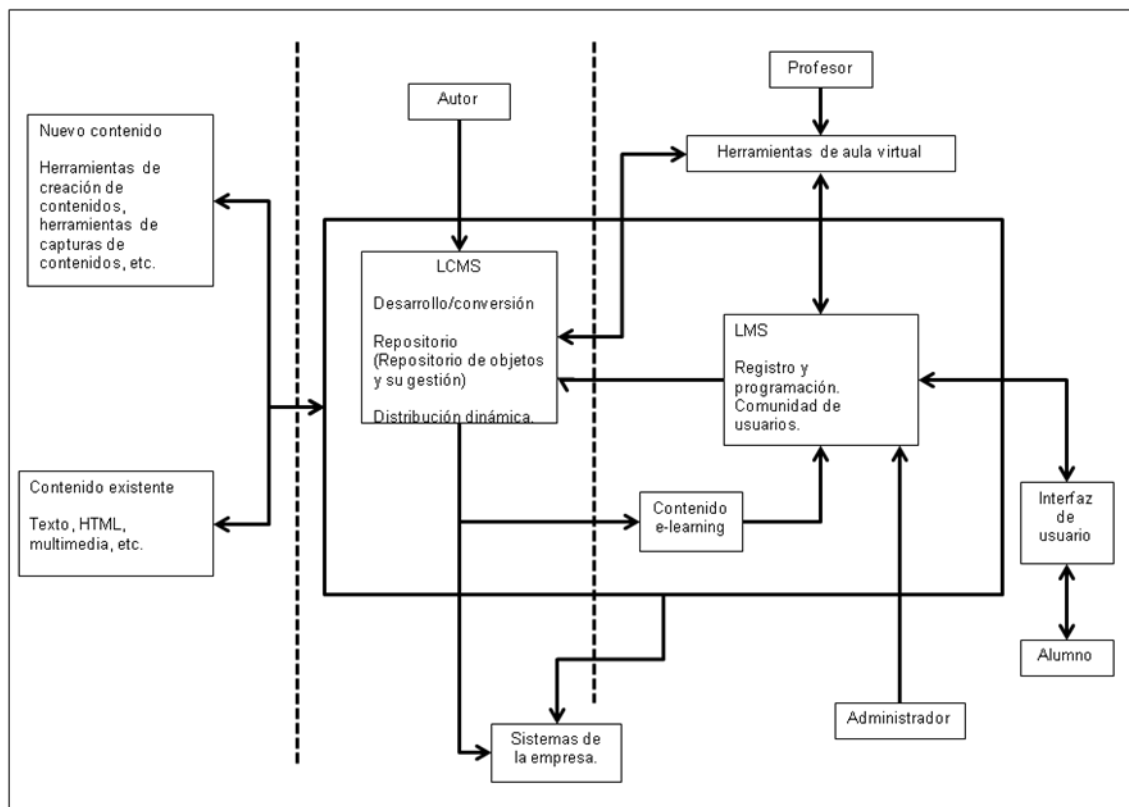
Con Internet se ha dado un nuevo paso en la educación a distancia convirtiéndola en una experiencia virtual. E-Learning es la forma de designar a este tipo de educación que se brinda, resultado de la combinación de tecnología, contenidos educativos, metodologías y personas.

La formación e-learning abre nuevas posibilidades y retos a la enseñanza gracias al uso combinado del ordenador y las redes de comunicación. El e-learning, entendido como aprendizaje a través de Internet o Intranet, también tiene otras denominaciones: tele-formación, formación online, *Web-based training*, *Web-based instruction*, *online learning*. Sea como fuere, el e-learning está posibilitando que los diseñadores de cursos tengan en su mano un abanico de posibilidades sin precedentes: formación a distancia tradicional, presencial, e-learning, semipresencial, etc.

No debe existir una confrontación entre el aprendizaje tradicional y el aprendizaje online, sino que ambos sistemas pueden y deben ser complementarios. Incluso, son frecuentes las acciones formativas en las que se combina con éxito, la formación presencial con e-learning. El Campus Virtual de la Universidad Complutense de Madrid es un ejemplo de este tipo de enseñanza.

Las principales ventajas que ofrece la educación virtual son (PHC Consultores 10): (i) reducción de costes, (ii) ahorro en la capacitación de empresas muy descentralizadas, (iii) flexibilidad horaria (factor de suma importancia); (iv) interacción en los cursos, despertando el interés del usuario.

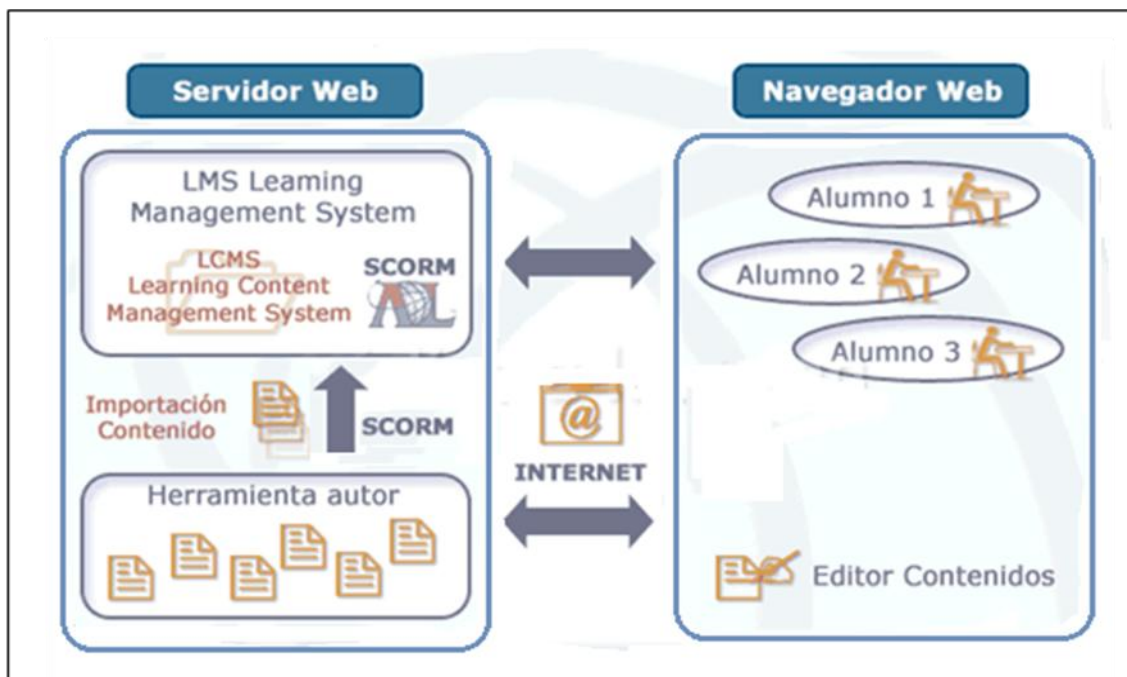
Una de las necesidades más evidentes a la hora de realizar una acción formativa mediante una metodología e-learning es la disposición de unos contenidos enfocados a las características de este medio.



**FIGURA 2.1 INTERACION DE USUARIOS CON LMS**

Como ya hemos comentado en la introducción, pieza clave en el e-learning, son los Sistemas de Gestión de Aprendizaje (LMS, *Learning Management System*). El LMS es una aplicación de servidor, generalmente web, el cual se emplea para administrar, distribuir y controlar las actividades e-learning. En la Figura 2.1 se describe brevemente la interacción de los usuarios con los LMS.

Las principales funciones del LMS son: gestionar usuarios, recursos y contenidos así como materiales y actividades para la formación o enseñanza de un material determinado; administrar el acceso, controlar y realizar un seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes de avances, y gestionar servicios de comunicación como foros de discusión y videoconferencias.



**FIGURA 2.2 LMS Y SU GESTION DE CONTENIDOS**

Por lo general, el LMS no incluye la posibilidad de crear sus propios contenidos, se focaliza en gestionarlos, representado en la Figura 2.2. La labor de creación de contenidos se desarrolla mediante los Sistemas de Gestión de Contenidos del Aprendizaje (LCMS, *Learning Content Management Systems*). El LCMS es un sistema independiente o integrado al LMS, que gestiona y administra los contenidos. Normalmente el LCMS utiliza el lenguaje XML, Extensible Markup Language (XML 10) y sigue los estándares de la enseñanza digital IMS, Innovation Adoption Learning (IMS 10) y SCORM (SCORM 10).

SCORM es un estándar promovido por ADL, Advanced Distributed Learning (ADL 10), que incluye un conjunto de especificaciones para la creación de objetos de aprendizaje reutilizables, haciendo a estos objetos compatibles con otros sistemas LMS.

Las especificaciones IMS son especificaciones basadas en tecnologías abiertas (XML) para facilitar las actividades de aprendizaje sobre tecnología web, principalmente para el intercambio de contenidos y de información sobre los estudiantes. Abarca la accesibilidad y adaptación del estudiante, la definición de competencias, el empaquetamiento de contenidos, información de agentes del proceso educativo, el diseño del aprendizaje a través de un lenguaje para expresar diferentes modelos pedagógicos, así como la formación de repositorios de contenidos digitales (IMS 05). El uso de estas especificaciones permite que la estructura de los contenidos de un curso sea compatible para todos los LMS que las utilicen.

Todas las ventajas ofrecidas por este conjunto de tecnologías, ofrecen, nuevos medios para llegar a un mercado más amplio, a universidades, instituciones de enseñanza superior, asociaciones y grupos de asesores, suscitando el uso de este nuevo tipo de enseñanza para sus programas educativos. En Estados Unidos existen excelentes programas en funcionamiento virtualmente en las principales universidades, como la Universidad de Phoenix. España se sitúa a la cabeza en ofrecer este tipo de educación, en español (E-learning 02).

Existe una amplia gama de LMS, que puede dividirse según su distribución, sistemas propietarios y sistemas de código abierto. Bajo la línea de sistemas propietarios, destacan BlackBoard, WebCT (adquirido por BlackBoard), Desire2Learn, eCollege, Saba Learning,

Kubbe, etc. Ganando una gran cuota del mercado, están los sistemas de código abierto, monopolizado por Moodle, y seguido por otras plataformas como Sakai, ATutor, Illias, Docebo, Dokeos, .LRN, Olat, o Claroline, entre otras. En la Figura 2.3 se muestra una comparativa de uso de algunas de estas plataformas.

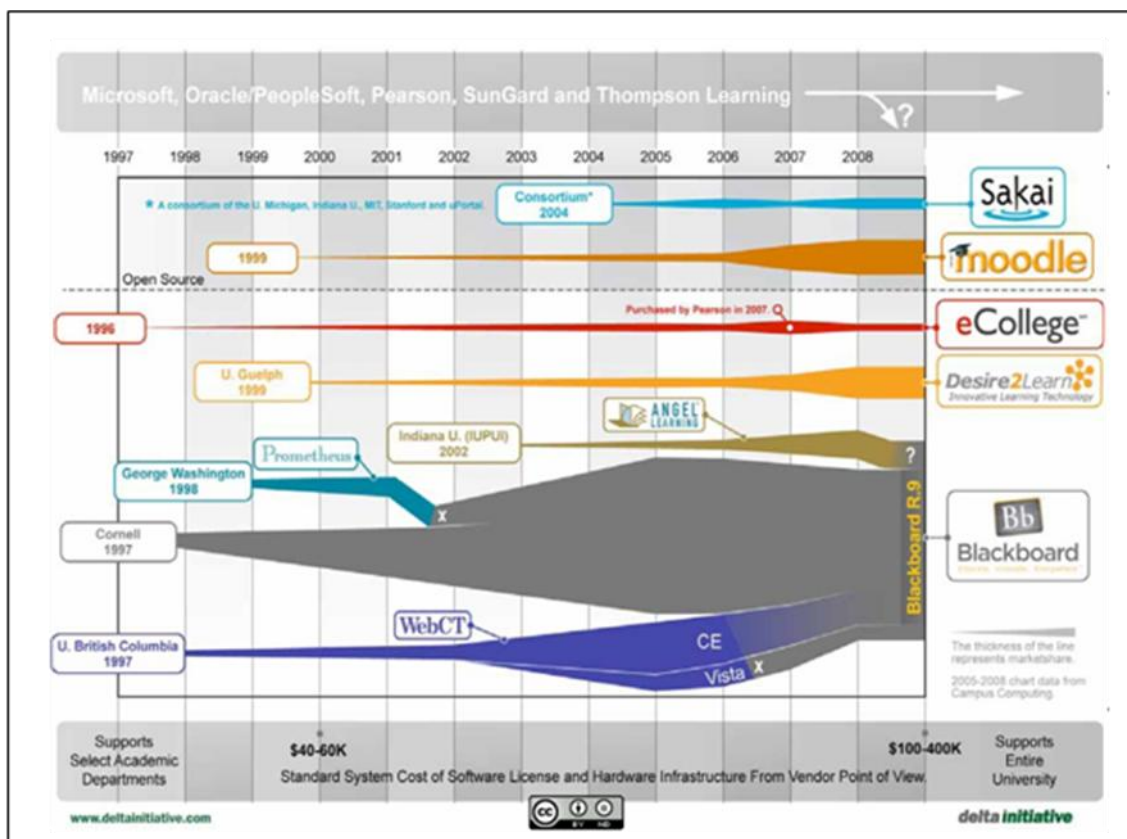


FIGURA 2.1 PLATAFORMA EN EL MERCADO ACTUAL

A continuación se describen algunas de las plataformas más relevantes. En particular nos centramos en aquellas plataformas que ya están incorporadas al Campus Virtual de la Universidad Complutense de Madrid y se espera que continúen durante los próximos años, y en aquellas que podrían llegar a formar parte de dicho Campus Virtual.

## 2.2 BLACKBOARD 9.1

Blackboard es una plataforma de educación electrónica que permite la administración de cursos para el aprendizaje (BlackBoard 10). Se trata de una plataforma de pago perteneciente a la empresa Blackboard con el mismo nombre, fundada en 1997. La plataforma se ha basado en la experiencia del cliente y ha ido mejorando gracias a la participación de los mismos.

Blackboard sirve para desarrollar cursos virtuales para impartir y distribuir contenidos en diversos formatos (texto, sonido, video y animación), realizar evaluaciones en línea, llevar a cabo el seguimiento académico de los estudiantes, asignar tareas y desarrollar actividades colaborativas mediante la participación en chats, foros y aulas virtuales. Ofrece herramientas para la interacción síncrona y asíncrona, la publicación de contenidos y posibilita la administración académica por parte de los profesores.

La versión más reciente de la plataforma Blackboard es la 9.1, también conocida como "Blackboard Learn", que salió al mercado en enero del 2009. De esta versión destaca que es más flexible y abierta, permitiendo que las instituciones puedan utilizar la plataforma como la



base para cualquier tecnología complementaria que deseen apoyar para su proceso de enseñanza o aprendizaje (BlackBoard 10).

Entre las características de esta nueva versión podemos destacar las siguientes (APB 09, BlackBoard DS 9.1, RJE 10):

1. Promueve la colaboración activa y el aprendizaje social gracias a la integración de “wikis” en cada asignatura, blogs y diarios que fomentan la comunicación y la reflexión, y la creación de herramientas de grupos que simplifican la comunicación y colaboración entre grupos.
2. Añade una nueva interfaz de usuario que permite crear y gestionar los contenidos de forma intuitiva basada en la experiencia de la Web 2.0. Además, la interfaz es personalizable por el propio usuario.
3. Permite la integración directa de contenidos multimedia de la Web 2.0, como videos de youtube.com, imágenes de flickr.com o presentaciones de slideshare.com.
4. Además de contar con un sistema de alertas y notificaciones, Blackboard cuenta con una aplicación para los dispositivos móviles proporcionando así una enseñanza y aprendizaje interactivos que se pueden seguir desde cualquier lugar.
5. Por otro lado, cuenta con un sistema de accesibilidad para las personas con discapacidades visuales certificado por la *National Federation of the Blind for Non-Visual Accessibility*.
6. Incorpora así mismo un sistema de mensajería instantánea para enriquecer la colaboración entre estudiantes y profesores.

Cada día, la plataforma Blackboard 9.1 es utilizada en miles de instituciones y por millones de personas. Blackboard es la principal plataforma de gestión de contenidos de aprendizaje y de participación de alumnos.

Según un informe realizado en colaboración con la Universidad de California, y como se puede observar en la figura 2.4 (Moodle vs Sakai 10), queda clara la hegemonía de Blackboard 9 sobre las soluciones de código abierto como por ejemplo Moodle o Sakai. No obstante, Blackboard ha ido perdiendo cuota de mercado paulatinamente en los últimos años.

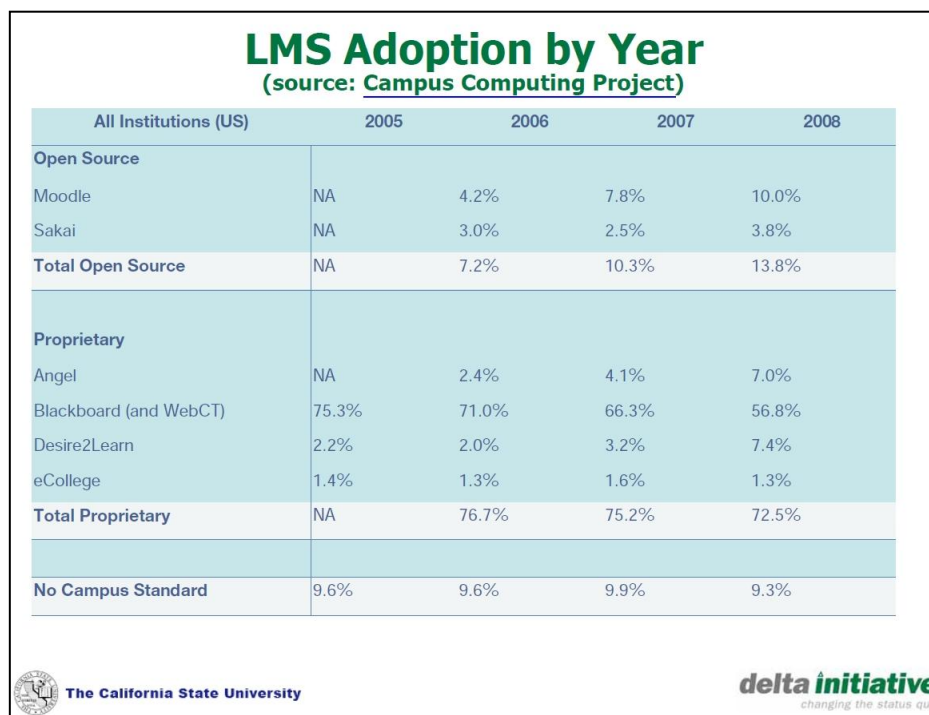


FIGURA 2.2 ESTADÍSTICAS DE USO DE LAS PLATAFORMAS

Al tratarse de una plataforma comercial, Blackboard no se ha incluido en la integración de plataformas y servicios de nuestro proyecto.

## 2.3 MOODLE 2.0

Moodle es la principal plataforma educativa virtual dentro de las de libre distribución. La palabra Moodle proviene del término anglosajón *Module Object-Oriented Dynamic Learning Environment*. Esta plataforma fue creada por Martin Dougiamas que trabajaba como administrador de WebCT para la universidad australiana de Curtin. Su intención fue la de dar un enfoque pedagógico diferente al que ofrecían el resto de plataformas, haciendo más hincapié en la contribución de los estudiantes a su propia educación (Moodle 10).

La primera versión de Moodle vio la luz el 20 de agosto de 2002, desde entonces han salido diferentes versiones, hasta 2.0 que se encuentra en fase Beta. Se espera que esta última versión esté terminada a finales de julio de 2010.

Las estadísticas proporcionadas por la Universidad Estatal de California y la compañía Delta Initiative, hablan de Moodle, con un 10% de la cuota total de mercado en el año 2009. Se sitúa así como la plataforma de software libre más utilizada, por encima de Sakai, con un 4%. En general Moodle solo es superada por Blackboard, que ocupa el 56% del mercado.

Según las estadísticas oficiales de Moodle.org actualmente existen 48.000 sitios validados, con un número total de usuarios registrados cercano a los 35 millones. Moodle está extendida en 210 países, y está traducido a más de 80 idiomas. Como vemos en la Tabla 1, el país con más sitios registrados son los Estados Unidos con 8.700, a continuación aparece España con 4.237, con lo que vemos la gran implantación que tiene la plataforma en nuestro país. Con estos datos el mapa mundial queda coloreado como vemos en la Figura 2.5.

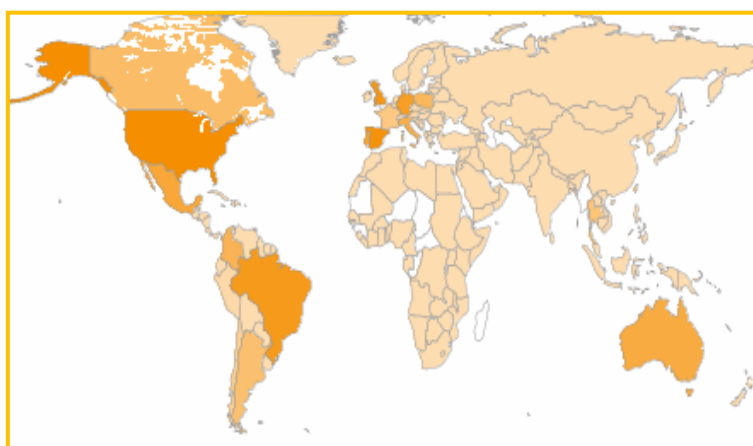


FIGURA 2. 3 UTILIZACION DE MOODLE POR PAISES.

País	Sitios
<b>EEUU</b>	8,700
<b>España</b>	4,237
<b>Reino Unido</b>	3,085
<b>Brasil</b>	3,055
<b>Alemania</b>	2,239
<b>Portugal</b>	1,747
<b>México</b>	1,663

TABLA 1. TOP 7 PAISES

Arquitectónicamente, Moodle es una aplicación web escrita en PHP por lo que se puede ejecutar en cualquier entorno que soporte este lenguaje interpretado, es decir, en prácticamente cualquier sistema operativo como son Unix, GNU/Linux, OpenSolaris, FreeBSD, Windows o Mac OS X. Este código en PHP está protegido por la licencia *General Public License* de GNU, la que permite su libre distribución, modificación y uso.

Los datos de la plataforma se almacenan en una única base de datos SQL. A partir de la versión 1.7 de Noviembre de 2006, se utiliza una abstracción total en esta base de datos, lo que permite a los administradores elegir entre varios sistemas de gestión de bases de datos. Algunos de estos son: MySQL, PostgreSQL, Oracle y Microsoft SQL server.

El código fuente de Moodle se desarrolla mediante un proyecto de software libre apoyado por un equipo de programadores y una gran comunidad internacional de usuarios que se encargan de desarrollar, traducir y solucionar fallos presentes en los fuentes la plataforma. Las actividades de esta comunidad se coordinan a través de los foros de la web principal de la plataforma. Asimismo en la misma web se dispone de un *tracker* para gestionar el control de versiones y las evoluciones en el código.

Para utilizar Moodle no es necesario el pago de ninguna cantidad de dinero, pero existe un sistema de donaciones donde los usuarios o instituciones pueden realizar sus aportaciones económicas para que el proyecto se mantenga actualizado y en constante progreso.

Para instalar Moodle son necesarias varias aplicaciones. Lo primero es un entorno donde se pueda ejecutar el código en PHP. También se necesitan una base de datos SQL y un gestor compatible. La última condición es un servidor de aplicaciones para desplegar la plataforma y que sea accesible a los usuarios.

La instalación de Moodle puede realizarse dos formas:

- Integrada: En esta instalación el servidor de aplicaciones y la base de datos se instalan en el mismo servidor físico. No es la recomendada para utilizarla en producción, pero es válida si lo que se desea es probar la plataforma y aprender su funcionamiento.
- Distribuida: En este caso el servidor de aplicaciones y el de la base de datos se encuentran en servidores físicos diferentes. Esta opción es la más recomendada por su seguridad y eficiencia.

En nuestro caso hemos utilizado la primera opción. En la propia web de la plataforma está disponible un paquete que nos permite este tipo de instalación. Este paquete contiene un servidor Apache, el gestor de bases de datos MySQL y el intérprete de PHP, además de los archivos fuente de la plataforma. Este paquete está disponible para su descarga gratuita en versiones para Windows y para Linux. Están disponibles todas las versiones que han ido saliendo incluida la versión 2.0.

Una de las principales características de Moodle es su interoperabilidad. La plataforma posee varios métodos para realizar la identificación de usuarios de manera automática, como son LDAP, Shibboleth o IMAP (EduTools 10). También se permite la creación de usuarios mediante la interacción con una base de datos externa o utilizando el estándar IMS Enterprise para la gestión mediante archivos XML. Incluso la gestión de los recursos de una asignatura se puede realizar mediante varios métodos estándar, como por ejemplo SCORM o IMS Content Packaging.

También se permite la importación de cursos desde otras plataformas como Blackboard. Estas características hacen de Moodle una plataforma única.

Sin embargo, Moodle al estar programado en PHP requiere un especial cuidado a la hora de configurar el servidor para ser usado por muchos usuarios. Otra de las desventajas de Moodle es su inexistente servicio técnico gratuito debido a que es una plataforma de código abierto y de libre distribución. Por esto hay que depender de empresas consultoras externas que den soporte para la plataforma.

## 2.4 SAKAI 2.6

La plataforma Sakai surge en el 2004 como una iniciativa para integrar funcionalidades de e-learning en un portal académico, creado por las principales instalaciones educativas del mundo, permitiendo la enseñanza de gran alcance, el aprendizaje y la colaboración en la investigación (UVG). Este portal integra un entorno virtual de enseñanza-aprendizaje para la educación compuesto por módulos, permitiendo el uso exclusivo de los módulos que se requieran. De este modo, se permite crear un ambiente de trabajo de acuerdo a las necesidades de cada curso, proyecto o investigación. Es una herramienta de apoyo que facilita procesos de comunicación y contacto entre profesores y alumnos, capaz de competir con sus equivalentes comerciales Blackboard / WebCT y que mejore otras iniciativas de código abierto como Moodle.

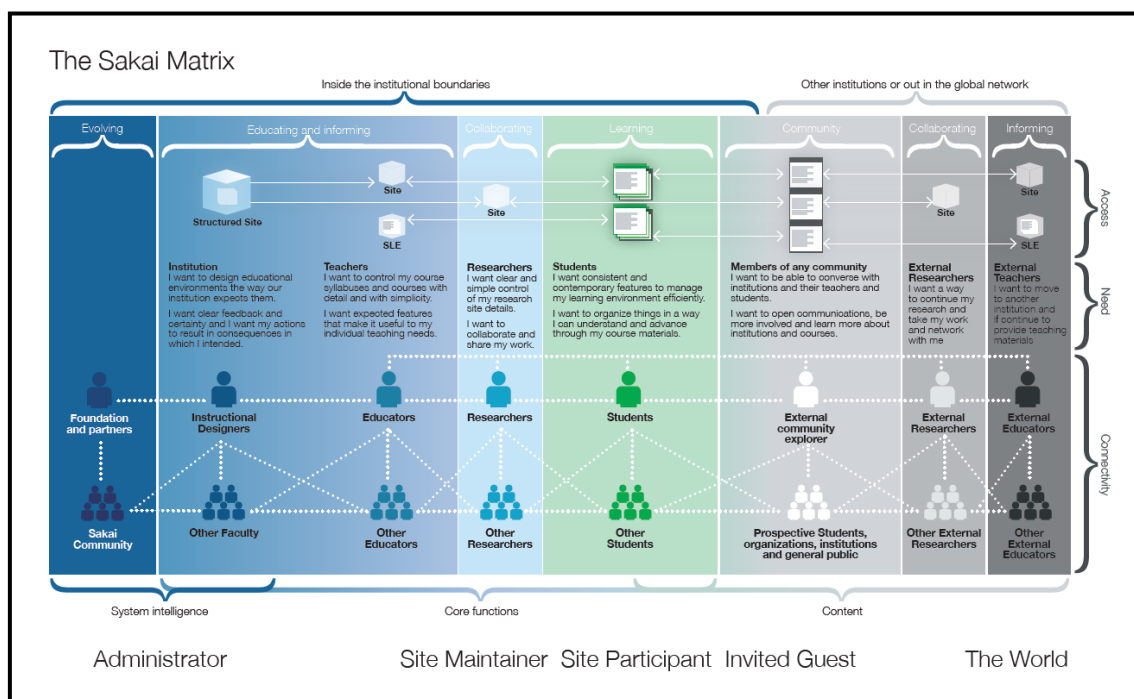


FIGURA 2.4 PLATAFORMA SAKAI Y SUS USUARIOS

Sakai se distribuye como software de código abierto y libre bajo la licencia de la Comunidad educativa. Este enfoque abierto de Sakai permite a las instituciones implementar, usar y modificar el software de la manera que sea más compatible con sus prácticas, necesidades y objetivos, es decir, permite el desarrollo de mejoras locales para ser incorporadas en versiones futuras del software.

Es una plataforma adaptable, interoperable, confiable y escalable para el aprendizaje y la colaboración como muestra la Figura 2.6. El sistema de roles y permisos de Sakai es especialmente flexible y potente. Para los cursos, Sakai proporciona características para complementar y mejorar la enseñanza y el aprendizaje. Para la colaboración, Sakai tiene herramientas para ayudar a organizar la comunicación y el trabajo colaborativo en el campus y en todo el mundo. Finalmente, para carteras, incluye la robusta cartera de código abierto Sistema e-Portafolio como una parte fundamental del software Sakai.

Sakai es líder del mercado en el suministro de características innovadoras. La comunidad de Sakai se beneficia de las funcionalidades que desarrollan el resto de organizaciones,

manteniendo la funcionalidad de Sakai en la vanguardia de la industria, convirtiendo a esta plataforma en un entorno muy potente (Sakai 10).

Una de las principales ventajas de la herramienta Sakai es su alta grado de personalización, satisfaciendo las necesidades de la institución en cuanto a su método de enseñanza.

Sakai es una aplicación empresarial altamente escalable, con instalaciones de producción total de más de 200.000 usuarios y 20.000 usuarios simultáneos. Esta configuración de Sakai para soportar esta escala de uso depende, en gran medida, de los detalles de su implementación y el uso de patrones.

Arquitectónicamente, Sakai se apoya en las más novedosas y robustas tecnologías. Desarrollado en Java, utiliza como servidor web Apache Tomcat y como framework de interfaz de usuario, JSF (Java Server Faces). La conexión con su base de datos coexiste con Hibernate, permitiendo un acceso rápido y escalable. En la capa de almacenamiento, permite el uso tanto de Oracle como MySQL, dependiendo de las necesidades de cada proyecto. Funciona de forma independiente de la plataforma de ejecución y actúa en los navegadores (Mozilla Firefox, Internet Explorer, Google Chrome, Sakafi, Opera,...) y sistemas operativos más populares (Windows, Linux, Mac,...)

Sin embargo, Sakai presenta diversos problemas. El primer punto negativo es la documentación. No por ser escasa sino por estar bastante desorganizada. Hay un gran volumen de información 100% colaborativa, tanto para desarrolladores como para usuarios, y resulta una tarea ardua encontrar lo importante y separarlo del resto. La página de inicio del proyecto Sakai muestra algunos enlaces útiles pero no suficientes. Otra desventaja añadida a la documentación, es la información obsoleta, bastante frecuente. Una posible solución a este problema es la publicación del libro oficial de Sakai que maquilla un poco este problema.

Su documentación puede dividirse en dos grandes grupos:

- Confluence: Wiki de la comunidad de Sakai, fomentando la colaboración de la comunidad en torno a la construcción, implementación y uso de Sakai (Confluence 10). Constituido por espacios de trabajo, documentación (información de versiones, documentación del código, prácticas habituales, etc.) y gestión y coordinación del proyecto (fechas importantes, resúmenes de estado, lista de proyecto, etc.).
- El sistema bugtracking, JIRA (poco intuitivo). Foro creado para facilitar la comunicación y colaboración en Sakai. De este modo, se realizan seguimientos de errores, de tareas e incluso, de peticiones de nuevas funcionalidades (Sakai 10). Resulta una herramienta sumamente útil pero tediosa ya que su nivel de organización no es óptimo, generando la necesidad de invertir gran cantidad de tiempo para cualquier consulta.

El segundo problema está en la magnitud curva de aprendizaje a la hora de desarrollar. La plataforma Sakai se basa en estándares tecnológicos abiertos muy avanzados, y debido a esto hay que emplear un gran número de horas hasta que se empieza a obtener resultados. Este hecho afecta negativamente a la comunidad de usuarios, resultando una comunidad bastante escasa.

Otro punto negativo de la plataforma es la manera de trabajar respecto a la gestión de calidad en la herramienta. Los encargados del repositorio oficial hacen comprobaciones exhaustivas de los posibles bugs de cada parche generado y aplicado sobre la rama inestable (*trunk*), lo cual mejora mucho la fiabilidad de los cambios. Pero también se traduce en un retraso bastante importante del arreglo más nimio sobre la aplicación en las ramas estables (*branches*). La

solución a este problema está en parchear manualmente nuestra instalación, lo cual es perfectamente válido, pero hace que el trabajo de mantenimiento se multiplique.

Por último, las versiones actuales no pueden considerarse *user friendly*, mostrando dificultades en cuanto a:

- La migración de contenidos (PDFs, DOCs,...). La necesidad de adaptación en la estructura de contenidos es un aspecto crítico. La solución a este problema se vería resuelto si el soporte de estándares como IMS CP fuera completo. Sin embargo, no todas las plataformas son compatibles con las mismas versiones de IMS CP.
- La configuración de los roles y permisos es muy flexible pero a la vez muy compleja. Cada uno de estos roles y permisos pueden variar su uso entre algunas versiones.
- Sakai tiene una gran variedad de herramientas que, en ocasiones, sirven para lo mismo.
  - Coexisten dos herramientas de exámenes (Samigo y Mneme)
  - Coexisten dos herramientas de blog (Blogger y Blogwow!),...

A pesar de estos inconvenientes, la plataforma Sakai se encuentra en uso en más de 200 universidades, colegios y escuelas de todo el mundo como se detalla en la Figura 2.7.



**FIGURA 2.5 USO DE SAKAI EN EL MUNDO**

Sakai 3 será la próxima nueva versión del software Sakai. Los usuarios finales de Sakai, que debido a su aumento de familiaridad con la tecnología Web 2.0, demandan un entorno más flexible y que permita un mayor control; el aumento exponencial del uso de redes sociales y la gran utilidad demostrada de Sakai en la investigación, son motivos suficientes para mejorar la

plataforma existente. En suma a esta necesidad, aparecen nuevos estándares y proyectos de código abierto que ofrecen una fácil integración a Sakai.

La finalidad de Sakai 3 no es sólo una mejora incremental de Sakai, crear una versión mejor y más económica que Blackboard; la finalidad es crear un sistema de colaboración académico diferente.

Se distinguen por tanto dos líneas de mejora; mejoras para usuarios y para desarrolladores, que son en definitiva, mejoras para la institución. Para el usuario; destacan las redes sociales y la creación y gestión de contenidos. Por su parte, los tecnólogos verán mejorado la escalabilidad y resistencia, una de las principales característica de la plataforma, así como también se facilitará la instalación y el mantenimiento.

## 2.5 COMPARACIÓN ENTRE PLATAFORMAS

Esta sección repasa las características fundamentales de cada plataforma para finalmente comparar las plataformas. Compararemos las funcionalidades de Sakai, Moodle y Blackboard.

Moodle y Sakai son sistemas de software libre por lo que se producen una serie de dificultades relacionadas con la evolución de las aplicaciones. En general el software propietario, como Blackboard, tiene una hoja de ruta mejor definida y los plazos se suelen cumplir en mayor medida que con este tipo de aplicaciones. Otro de los problemas que aparecen con las plataformas de software libre es la pobre y desorganizada documentación que existe sobre ellas. El soporte técnico que proporciona Blackboard hace que muchas instituciones se decanten por este sistema, mientras que es una de los mayores problemas en Moodle y Sakai.

A la hora de la instalación existen grandes diferencias entre Moodle y Sakai. La primera tiene una instalación bastante sencilla y hay paquetes disponibles con todas las herramientas necesarias, que permiten desplegar la plataforma en pocos minutos. Por el contrario la instalación de Sakai conlleva un proceso que, dependiendo de la habilidad y los conocimientos del instalador, puede llegar a ser complicado y tedioso.

Las plataformas de software libre están implementadas bajo tecnologías diferentes, lo que le aportan características particulares. Sakai esta implementado en Java, utilizando el Framework Java Server Faces (JSF) lo que le da robustez a la aplicación y una gran solidez a la hora de manejar muchos usuarios. Por el contrario, para los desarrolladores es más complicado entender el funcionamiento interno de la plataforma y lleva un tiempo de aprendizaje. Por otro lado Moodle esta implementado en PHP lo que facilita su despliegue y su adaptabilidad para multitud de servidores web. El uso de PHP provoca que tengamos que cuidar especialmente los accesos concurrentes de muchos usuarios.

Las principales características de las tres plataformas están resumidas a continuación (EduTools 10):

2010 EduTools / WCET			
Plataforma	Blackboard	Sakai	Moodle
<b>Herramientas de Comunicación</b>			
<b>Foros de Discusión</b>	Corrector ortográfico. Ordenación de los Temas por fecha, hilo o autor. Expansión y contracción de conversaciones. Búsqueda de mensajes. Los mensajes pueden	Corrector ortográfico. Posibilidad de enviar mensajes mediante email. Recepciones de emails diarios con resúmenes o temas enteros.	Corrector ortográfico. Posibilidad de enviar mensajes mediante email. Recepciones de emails diarios con resúmenes o temas enteros. Subscripción al foro

	contener URLs, archivos adjuntos y código HTML. Editor de ecuaciones.		mediante RSS.
<b>Gestión de Recursos</b>	Envío de archivos al profesor. Los alumnos pueden compartir sus carpetas personales con otros alumnos. Carpetas privadas para cada alumno. Carpetas compartidas para grupos de alumnos. Los profesores pueden enviar archivos a las carpetas privadas de los alumnos.	Envío de archivos al profesor. Los alumnos pueden crear sitios para proyectos con otros alumnos.	Envío de archivos al profesor.
<b>Envío de Mensajes</b>	Envío de mensajes mediante la herramienta interna o mediante la dirección de email. Profesores pueden mandar mensajes a toda la clase a la vez. Se permiten archivos adjuntos en los mensajes.	Envío de mensajes mediante la herramienta interna. Profesores pueden mandar mensajes a toda la clase a la vez.	Envío de mensajes mediante la herramienta interna. Existe un libro de direcciones de los alumnos. Reenvío de los mensajes a direcciones externas.
<b>Diario de Clase / Notas</b>	Los alumnos pueden crear notas privadas sobre cada asignatura.	Los alumnos pueden crear un directorio de notas para cada Asignatura.	
<b>Chat</b>	Conversaciones de grupos ilimitadas. Los profesores pueden hacer de moderadores de las salas de chat y bloquear usuarios. Generación de Logs.	Conversaciones de grupos limitadas. Generación de Logs. Los profesores pueden nuevas salas de chat. Lista de usuarios en cada sala.	Conversaciones de grupos limitadas. Generación de Logs. Los profesores y los alumnos pueden nuevas salas de chat. Los profesores pueden hacer de moderadores de las salas de chat y bloquear usuarios.
<b>Pizarra Online</b>	Subida de imágenes y presentaciones PowerPoint. Soporte para símbolos matemáticos. Grabación de las sesiones de pizarra.	Necesaria la integración de una pizarra externa.	Necesaria la integración de una pizarra externa.
<b>Herramientas de Productividad</b>			
<b>Calendario</b>	Los alumnos y profesores pueden crear eventos en el calendario de cada asignatura.	Los alumnos y profesores pueden crear eventos en el calendario de cada asignatura. Posibilidad de subscripción RSS para alertar sobre cambios en el calendario.	Los alumnos y profesores pueden crear eventos en el calendario de cada asignatura. Posibilidad de subscripción RSS para alertar sobre cambios en el calendario.



<b>Trabajo offline</b>	Los usuarios pueden descargar el contenido de una asignatura en un formato que puede ser almacenado y leído de forma local.	Los usuarios pueden descargar el contenido de una asignatura en un formato que puede ser almacenado y leído de forma local. Los profesores pueden descargarse los trabajos recibidos de los alumnos para evaluarlos posteriormente.	
<b>Ayuda</b>	El sistema incluye una ayuda para los alumnos que explica cómo utilizarlo.	El sistema incluye una ayuda para los alumnos que explica cómo utilizarlo. Cada herramienta tiene su propio manual de uso.	El sistema incluye una ayuda para los alumnos que explica cómo utilizarlo. Cada herramienta tiene su propio manual de uso.
<b>Herramientas de trabajo para alumnos</b>			
<b>Grupos de trabajo</b>	Los profesores pueden crear grupos de trabajo. El sistema puede crear grupos aleatorios de un tamaño determinado. Cada grupo dispone de su propio foro, chat y pizarra. Cada grupo puede tener asignados diferentes trabajos. Los grupos pueden ser privados.	Los profesores pueden crear grupos de trabajo. Cada grupo puede tener asignados diferentes trabajos. Los grupos pueden ser privados.	Los profesores pueden crear grupos de trabajo. El sistema puede crear grupos aleatorios de un tamaño determinado. Los alumnos pueden seleccionar su grupo. Cada grupo dispone de su propio foro, chat y pizarra. Cada grupo puede tener asignados diferentes trabajos. Los grupos pueden ser privados.
<b>Trabajo Comunitario</b>	Los alumnos pueden crear grupos de estudio, de interés o clubs. Los alumnos de diferentes asignaturas pueden interactuar en foros y chats globales.	Los alumnos pueden crear grupos de estudio, de interés o clubs.	Los alumnos pueden crear grupos de estudio, de interés o clubs. Los alumnos de diferentes asignaturas pueden interactuar en foros y chats globales.
<b>Herramientas de Administración</b>			
<b>Autenticación</b>	Se puede permitir la entrada a invitados a todas las asignaturas. Sistemas de autenticación soportados: <ul style="list-style-type: none"> <li>• Servidor LDAP.</li> <li>• Protocolo Kerberos.</li> <li>• Shibboleth.</li> <li>• Central Authentication Service (CAS).</li> <li>• IMAP, POP y NNTP.</li> </ul>	Se puede permitir la entrada a invitados a todas las asignaturas. Sistemas de autenticación soportados: <ul style="list-style-type: none"> <li>• Servidor LDAP.</li> <li>• Protocolo Kerberos.</li> <li>• Shibboleth.</li> <li>• Central Authentication Service (CAS).</li> </ul>	Se puede permitir la entrada a invitados a todas las asignaturas. Sistemas de autenticación soportados: <ul style="list-style-type: none"> <li>• Servidor LDAP.</li> <li>• Protocolo Kerberos.</li> <li>• Shibboleth.</li> <li>• Central Authentication Service (CAS).</li> <li>• IMAP, POP y</li> </ul>

	Se puede configurar un segundo método de autenticación en caso de que el principal falle.		NNTP.  Se puede configurar un segundo método de autenticación en caso de que el principal falle.
<b>Roles y Permisos</b>	Los roles pueden ser personalizados. El sistema permite restricciones de acceso basándose en los roles. Se permiten múltiples roles en la misma asignatura.	Los roles pueden ser personalizados. El sistema permite restricciones de acceso basándose en los roles. Se permiten múltiples roles en la misma asignatura.	Los roles pueden ser personalizados. Se pueden crear infinitud de roles y definir sus permisos. El sistema permite restricciones de acceso basándose en los roles. Se permiten múltiples roles en la misma asignatura.
<b>Alta de usuarios</b>	Los profesores añaden a los alumnos manualmente. También se permite el alta por parte del alumno. Exportación e importación e alumnos mediante archivos de texto con los campos bien delimitados.	Los profesores añaden a los alumnos manualmente. También se permite el alta por parte del alumno. Exportación e importación e alumnos mediante archivos de texto con los campos bien delimitados.	Los profesores añaden a los alumnos manualmente. También se permite el alta por parte del alumno. Exportación e importación e alumnos mediante archivos de texto con los campos bien delimitados. Compatibilidad de transferencia de datos mediante el estándar IMS Enterprise Specification v1.1 utilizando servicios web.
<b>Herramientas de gestión de Cursos</b>			
<b>Tipos de exámenes</b>	Múltiples respuestas. Exámenes de Relacionar. Ordenar. Calcular. Rellenar los espacios. Respuesta corta. Encuestas. Ensayos. Exámenes con imágenes, videos y audio.	Múltiples respuestas. Exámenes de Relacionar. Calcular. Rellenar los espacios. Respuesta corta. Encuestas. Ensayos. Exámenes con imágenes, videos y audio.	Múltiples respuestas. Exámenes de Relacionar. Ordenar. Calcular. Rellenar los espacios. Respuesta corta. Encuestas. Ensayos. Exámenes con imágenes, videos y audio. Preguntas personalizadas.
<b>Gestión de exámenes</b>	El sistema puede generar test aleatorios. El profesor puede crear auto-evaluaciones. Límites de tiempo para un examen. Se permiten múltiples intentos de examen. Revisión de intentos de	El sistema puede generar test aleatorios. El profesor puede crear auto-evaluaciones. Límites de tiempo para un examen. Se permiten múltiples intentos de examen. Revisión de intentos de	El sistema puede generar test aleatorios. El profesor puede crear auto-evaluaciones. Límites de tiempo para un examen. Se permiten múltiples intentos de examen. Revisión de intentos de

	examen anteriores. Editor de ecuaciones para preguntas y respuestas.	examen anteriores.	examen anteriores. Editor de ecuaciones para preguntas y respuestas.
<b>Notas</b>	Tabla de notas automática cuando se crea un trabajo online. Los profesores pueden añadir notas para trabajos offline. Columnas de la tabla de notas personalizables. Exportación de tablas de notas. Escala de puntuación elegibles (Porcentajes, letras o aprobado/suspenso).	Tabla de notas automática cuando se crea un trabajo online. Los profesores pueden añadir notas para trabajos offline. Exportación de tablas de notas. Escala de puntuación elegibles (Porcentajes, letras o aprobado/suspenso).	Tabla de notas automática cuando se crea un trabajo online. Los profesores pueden añadir notas para trabajos offline. Columnas de la tabla de notas personalizables. Exportación de tablas de notas. Escala de puntuación elegibles (Porcentajes, letras o aprobado/suspenso).
<b>Gestión de cursos</b>	Los profesores pueden generar trabajos, evaluaciones y anuncios en función de fechas de inicio y fin. Contenidos pueden bloquear el curso a los estudiantes hasta que sean completados. Los profesores pueden relacionar un foro con un evento o una fecha determinada. Los profesores pueden restringir el acceso a un contenido a un determinado grupo de alumnos. Los profesores pueden determinar el acceso a un contenido basándose en los resultados de una actividad anterior o resultados de un alumno.	Los profesores pueden generar trabajos, evaluaciones y anuncios en función de fechas de inicio y fin. Los profesores pueden restringir el acceso a un contenido a un determinado grupo de alumnos.	Los profesores pueden generar trabajos, evaluaciones y anuncios en función de fechas de inicio y fin. Los profesores pueden restringir el acceso a un contenido a un determinado grupo de alumnos.
<b>Seguimiento de Alumnos</b>	Los profesores pueden observar la frecuencia y la duración de los accesos de los alumnos a los contenidos de las asignaturas. Los profesores tienen informes que indican el número de veces, la hora, la fecha, la frecuencia y la dirección IP de los accesos de los alumnos a los	La herramienta de estadísticas está en proceso de desarrollo.	Los profesores pueden observar la frecuencia y la duración de los accesos de los alumnos a los contenidos de las asignaturas. Los profesores tienen informes que indican el número de veces, la hora, la fecha, la frecuencia y la dirección IP de los accesos de los alumnos a los

	contenidos y módulos de la asignatura.		contenidos y módulos de la asignatura. Los profesores pueden comprobar los registros de navegación dentro de la asignatura de cada alumno.
<b>Requisitos de Software</b>			
<b>Base de Datos</b>	Oracle Microsoft SQL Server	Oracle Microsoft SQL Server Solo se requiere una BBDD y puede coexistir con tablas de otras aplicaciones.	Oracle Microsoft SQL Server MySQL PostGreSQL Solo se requiere una BBDD y puede coexistir con tablas de otras aplicaciones.
<b>Servidor</b>	Servidor Unix. Servidor Windows.	Servidor Unix. Servidor Windows.	Servidor Unix. Servidor Windows.

**TABLA 2. COMPARATIVA DE PLATAFORMAS**

## 3. ARQUITECTURAS ORIENTADAS A SERVICIOS

### 3.1 INTRODUCCIÓN

La Arquitectura Orientada a Servicios (en inglés *Service Oriented Architecture*, SOA), es un concepto de arquitectura de software que permite la creación de sistemas altamente escalables. Esta arquitectura, a su vez, brinda una forma bien definida de exposición e invocación de servicios colaborativos (comúnmente pero no exclusivamente servicios web), facilitando la interacción entre diferentes sistemas propios o de terceros mediante Internet y la Web (SOA JavaEE 09).

El concepto SOA presupone el nuevo paradigma en la implementación de arquitecturas de software. SOA es un modelo conceptual implementado bajo un conjunto de estándares, definidos mediante los principales organismos internacionales en el desarrollo de estándares tecnológicos, como son W3C, OASIS, WSI, etc (Grupo CMC 10).

La característica principal es que SOA ha conseguido la estandarización de la información en su contenido y estructura. SOA se basa en servicios de aplicación atómicos o débilmente acoplados con alta interoperabilidad alcanzada mediante la estandarización. En lugar de definir una API, SOA define la interfaz en términos de los protocolos y la funcionalidad. Estas características convierten a SOA en la metodología más apropiada para la integración y consolidación de aplicaciones distribuidas, haciendo posible que diferentes aplicaciones con diferentes lenguajes de programación servidas en diferentes máquinas puedan interoperar entre sí.

El elemento básico de SOA son los servicios. Un servicio es un componente software reutilizable que implementa una función de negocio con las siguientes características:

- No tiene estado: los servicios no son dependientes del estado de otros.
- Es auto-contenido: no contiene llamadas embebidas a otros servicios.
- Es accesible a través de una interfaz bien definida.
- Su interfaz se describe en un documento.

SOA define las siguientes capas de software:

- Aplicaciones básicas: sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades, donde las funcionalidades de la capa de aplicación son expuestas en forma de servicios (servicios web).
- De integración de servicios, que facilitan el intercambio de datos entre elementos de la capa de aplicación orientada a procesos empresariales internos o en colaboración.
- De composición de procesos, que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.
- De entrega, donde los servicios son desplegados a los usuarios finales.

Las ventajas que ofrece una arquitectura SOA son las siguientes:

- El bajo acoplamiento de los componentes entre sí es la característica clave que nos proporciona reutilización de componentes, escalabilidad y flexibilidad.
- Interoperabilidad: ya que esta arquitectura es independiente de la plataforma, del sistema operativo, del lenguaje de programación, etc.
- Por las anteriores características, nos facilita un rápido desarrollo y nos facilita el mantenimiento de nuestras aplicaciones.

- Aplicaciones más seguras y manejables: Las soluciones orientadas a servicios proporcionan una infraestructura común (y una documentación común también) para desarrollar servicios seguros, predecibles y gestionables. Conforme van evolucionando las necesidades de negocio, SOA facilita la posibilidad de añadir nuevos servicios y funcionalidades para gestionar los procesos de negocio crítico. Se accede a los servicios y no a las aplicaciones (Real World SOA 10).

Por el contrario, la principal desventaja de esta arquitectura es el rendimiento, si se compara con otras arquitecturas de computación distribuida como RMI, CORBA o DCOM, debido a la adopción de XML como base para la comunicación.

La implementación de la arquitectura SOA más extendida se realiza mediante los servicios Web.

Los Servicios Web son un sistema de software diseñados para permitir una interacción interoperable de máquina a máquina en una red. Cuentan con una interfaz descrita en formato procesable por las máquinas, concretamente, esa interfaz es WSDL. El resto de sistemas interaccionan con el Servicio Web mediante mensajes de SOAP y usando HTTP con serialización XML.

Los Servicios Web surgieron ante una necesidad de estandarizar la comunicación entre distintas plataformas (PC, Mainframe, Mac, etc.) y lenguajes de programación (PHP, C#, Java, etc.).

Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito- como por ejemplo DCOM y CORBA- por ser dependientes de la implementación del vendedor.

Otro gran problema es que se hacía uso de RPC (*Remote Procedure Call*) para realizar la comunicación entre diferentes nodos. Esto, además de presentar ciertos problemas de seguridad, tenía la desventaja de que su implementación era casi imposible en un ambiente como es Internet, pues muchos firewalls bloqueaban este tipo de mensajes, lo que hacía prácticamente imposible que dos computadoras estuvieran conectadas por Internet para comunicarse.

Los Servicios Web surgieron para poder lograr la comunicación entre diferentes plataformas. Por ello, en 1999 se comenzó a plantear el nuevo estándar, que terminaría utilizando XML, SOAP, WSDL, y UDDI (Historia WS 05), y que más adelante se detallan.

El uso de arquitecturas orientadas a servicios no implican únicamente el uso de servicios Web, pero nos centramos en estos últimos al ser los que hemos desarrollados para el objetivo del proyecto.

## 3.2 PRINCIPALES ESTÁNDARES SOA

### 3.2.1 XML

El estándar XML es el utilizado para la representación estructurada de los datos que envían y reciben los servicios web, y también para la creación de etiquetas. Es un estándar definido por W3C.

### 3.2.2 SOAP

Para que los servicios puedan comunicarse entre ellos, los mensajes deben ser codificados de acuerdo a las especificaciones SOAP 1.1 y SOAP 1.2. SOAP es el acrónimo de *Simple Object*

*Access Protocol*, que permite el intercambio de información estructurada en la implementación de Servicios Web, y que es independiente de la plataforma y el lenguaje. SOAP está basado en XML y es la base principal de los servicios web.

SOAP puede usar múltiples medios de transporte, como son SMTP, HTTP/HTTPS, FTP, etc. Se utiliza principalmente HTTP por ser un protocolo ampliamente difundido y que se encuentra menos restringido por firewalls, que generalmente bloquea puertos como el FTP, siendo bastante poco probable que lo esté el puerto HTTP. Además, SOAP utiliza XML como formato del mensaje, es decir, los mensajes SOAP son básicamente documentos XML que son intercambiados a través de la red. Además, puede combinar múltiples estándares que abarcan otras facetas: seguridad, confiabilidad, transacciones, etc.

### 3.2.3 WSDL

Para describir los mensajes y protocolos usados por los servicios web se utilizan los metadatos estándar, que son los utilizados por las aplicaciones y las infraestructuras para garantizar que los servicios puedan interoperar entre ellos. El más importante y el que hemos utilizado en este proyecto es WSDL. WSDL es un protocolo basado en XML que describe el mensaje que un servicio web puede recibir y enviar, indicando cuáles son las interfaces del servicio web junto al tipo de datos necesario para la utilización del mismo. Es el lenguaje más básico utilizado para describir la funcionalidad ofrecida por un servicio.

En el siguiente ejemplo de nuestro proyecto podemos observar la utilización de los anteriores estándares:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://localhost/WS/MoodleWSAnuncios.php"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://localhost/WS/MoodleWSAnuncios.php">

  <types>
    <xsd:schema targetNamespace="http://localhost/WS/MoodleWSAnuncios.php"
    >
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    </xsd:schema>
  </types>
  <message name="addAnuncioRequest">
    <part name="courseid" type="xsd:string" />
    <part name="titulo" type="xsd:string" />
    <part name="texto" type="xsd:string" />
    <part name="sesion" type="xsd:string" /></message>
  <message name="addAnuncioResponse">
    <part name="return" type="xsd:string" /></message>

  <portType name="MoodleWSAnunciosPortType">
    <operation name="addAnuncio">
      <input message="tns:addAnuncioRequest"/>
      <output message="tns:addAnuncioResponse"/>
    </operation>
  </portType>
</definitions>
```

```

</portType>
<binding name="MoodleWSAnunciosBinding" type="tns:MoodleWSAnunciosPortType">

  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

  <operation name="addAnuncio">
    <soap:operation soapAction="http://localhost/WS/MoodleWSAnuncios.php/addAnuncio" style="rpc"/>

    <input><soap:body use="literal" namespace="http://localhost/WS/MoodleWSAnuncios.php"/></input>

    <output><soap:body use="literal" namespace="http://localhost/WS/MoodleWSAnuncios.php"/></output>

  </operation>

</binding>
<service name="MoodleWSAnuncios">
  <port name="MoodleWSAnunciosPort" binding="tns:MoodleWSAnunciosBinding">
    <soap:address location="http://localhost/WS/MoodleWSAnuncios.php"/>
  </port>
</service>
</definitions>

```

Como vemos, se ha utilizado XML para la representación de los datos, SOAP para el intercambio del mensaje sobre HTTP, y WSDL para la descripción de la funcionalidad que ofrece, en este caso, la agregación de un anuncio a la plataforma.

### 3.2.4 UDDI

Es un modelo de directorios para servicios web. Es una especificación para mantener directorios estandarizados de información acerca de los servicios Web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los servicios web. Es un lugar en el cual podemos buscar cuales son los servicios web disponibles, una especie de directorio en el cual podemos encontrar los servicios Web publicados y publicar los servicios web que desarrollemos.

Debido a que los servicios web desarrollados en nuestro proyecto son utilizados por clientes que conocían de antemano qué servicios Web invocar, en este proyecto no ha sido necesaria la utilización del protocolo UDDI.

### 3.2.5 WS-SECURITY

Otro importante componente de los servicios web sobre todo para las aplicaciones empresariales es la Seguridad. WS-Security es un protocolo que Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

Los mensajes SOAP son protegidos vía los estándares de “WS-Security”, los cuales son ahora utilizados como medio en la definición de la manera en que la encriptación, autenticación y firmas digitales serán utilizados como mecanismos para asegurar las comunicaciones.

La implementación de los servicios web se puede realizar sobre tecnologías JAVA o .NET, o incluso otras (C, C++, etc.) utilizando los estándares definidos, integrándose cada capa de la arquitectura con la subyacente y exponiendo los servicios hacia la capa superior mediante WSDL, SOAP, http y su gestión sobre una catalogación UDDI.

## 3.3 IMPLEMENTACIONES J2EE

La plataforma JavaEE incluye una especificación de servicios web interoperables compuesta por tecnologías integradas, tales como la API de Java para Servicios Web bajo XML (JAX-WS),



la arquitectura Java para la vinculación de servicios web (JAXB) y SAAJ. También existe el estándar JAX-RPC para la creación de servicios Web; éste es el utilizado en J2EE 1.4, mientras que JAX-WS, está diseñado para las nuevas especificaciones de JavaEE y elimina mucha de la complejidad asociada con el desarrollo de servicios web. Esto se debe a que JAX-WS provee un modelo simple para la implementación de la lógica del negocio y exponen el “intercambio” como una interfaz WSDL utilizando anotaciones en el código implementado. Con JAX-WS, la complejidad de WSDL y el manejo de los mensajes SOAP se pueden ocultar por completo. Además, JAX-WS está basado por defecto sobre SOAP 1.1 y HTTP, aunque se pueden especificar otros protocolos de transporte. Éste estándar, JAX-WS, es por tanto el que hemos utilizado para la implementación de nuestros servicios web en Java (SOA JavaEE 09, Real World SOA).

Por otro lado, los servicios web pueden ser definidos bien como componentes de la capa web (como ha sido en nuestro caso), o bien como Enterprise JavaBeans (EJB's). El modelo de programación es similar en ambos casos.

Para proveer una máxima interoperabilidad, la plataforma Java proporciona soporte para los estándares de SOAP, WSDL y la totalidad de los esquemas XML entre otros.

El formato de entrada y salida de los mensajes requeridos interactuar con el servicio web se encuentran definidos en un documento WSDL. Para crear o acceder a un servicio web utilizando la tecnología JavaEE, es necesario mapear la información contenida en el fichero WSDL con sus correspondientes estructuras en el lenguaje Java. Esto es manejado de forma transparente por las tecnologías JAX-WS y JAXB. Gracias a ello, en la mayoría de los casos, no es necesario centrarse en el manejo de los documentos WSDL (SOA JavaEE 09).

A continuación mostramos como ejemplo el desarrollo de la operación de *login* implementada en nuestro servicio web en J2EE:

```
@WebMethod(operationName = "login")
public String login(@WebParam(name = "nombre")
String nombre, @WebParam(name = "clave")
String clave) {
    String resultado = null;
    try {
        Service service = new Service();
        Call nc = (Call) service.createCall();
        nc.setTargetEndpointAddress("http://localhost:8080/sakai-axis/SakaiLogin.jws?wsdl");
        nc.removeAllParameters();
        nc.setOperationName("login");
        nc.addParameter("nombre",XMLType.XSD_STRING, ParameterMode.IN);
        nc.addParameter("clave",XMLType.XSD_STRING, ParameterMode.IN);
        nc.setReturnType(XMLType.XSD_STRING);
        resultado = (String) nc.invoke(new Object [] { nombre, clave });
        System.out.println(resultado);
    } catch (Exception e) {
        System.out.print("Error: "+e.getMessage());
        resultado="Error: "+e.getMessage();
    }
    return resultado;
}
```

La implementación de la función es sencilla, siendo la única tarea del desarrollador la implementación de la misma. En este caso, observamos que este servicio web interactúa con

otro servicio web de Sakai, que también hemos desarrollado, siendo éste el que realmente realiza el acceso a la API de Sakai.

### 3.4 IMPLEMENTACIONES PHP

Puesto que la plataforma Moodle ha sido desarrollada en PHP, se ha decidido que los Servicios Web que interactuarán con la misma se implementen en PHP por mayor facilidad. Para ello, ha sido necesaria la utilización de NuSoap, un kit de herramientas que permite desarrollar Servicios Web en PHP mediante una serie de clases que lo componen. NuSoap está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1, y destaca por no requerir de módulos adicionales y de su fácil instalación y uso, pues solo es necesario descargar el paquete de NuSoap y colocarlo en el servidor web (AOS PHP).

Los servicios web que acceden a la plataforma Moodle están desplegados en la misma máquina, puesto que necesitan acceder a las API's de la plataforma. Sin embargo, la aplicación Web del Campus Virtual que consume estos servicios puede estar en otra máquina diferente, y desarrollada en Java. Vemos aquí cómo estos servicios web son accesibles desde otros entornos al estar estandarizados mediante SOAP, WSDL y XML.

NuSoap no es el único soporte para Servicios Web en PHP, pero sí el que está en una fase de desarrollo mucho más avanzada. Sin ir más lejos, PHP a partir de la versión 5 proporciona soporte para SOAP, pero está aun en una fase experimental.

A continuación mostramos como ejemplo el desarrollo de la operación de *login* implementada en nuestro servicio web en PHP:

```
//registro de la función
$server->register(
    'login',                // method name
    array('nombre' => 'xsd:string','clave' => 'xsd:string' ),    // input parameters
    array('return' => 'xsd:string'),    // output parameters
    'http://localhost/WS/MoodleWSUsuarios.php',    // namespace
    false,                // SOAPAction
    false,                // style
    'literal'              // use
);

// Definición el método como una función PHP
/**
 * Login en la plataforma de Moodle
 *
 * @param string $nombre
 * @param string $clave
 * @return boolean con el resultado del login
 */
function login($nombre,$clave) {
    //HTTPS is potentially required in this page
    httpsrequired();

    $auth=authenticate_user_login($nombre, $clave);
    $result="";
    if ($auth){
        //$result="Login success";
        $result=$auth->id;
        update_login_count();
        complete_user_login($auth);
    }
}
```

```
else $result="Login KO. Error";  
return new soapval('return', 'xsd:string', $result);  
};
```

Como podemos observar, distinguimos dos partes claramente diferenciadas: la declaración de la función como método del servicio web, definiendo su interfaz, y la implementación de la misma en lenguaje PHP.

### 3.5 SERVICIOS WEB J2EE FRENTE A SERVICIOS WEB PHP

A la hora de desarrollar los servicios web, podemos mencionar la sencillez que nos ha aportado el framework de J2EE, que facilita y simplifica su implementación, contando con el soporte de sus librerías (JAX-WS). Además, es muy sencillo su despliegue en la mayoría de los servidores de aplicaciones.

Sin embargo, para crear los servicios web en PHP nos hemos encontrado con más dificultades. La primera de ellas fue la obtención de la librería *nusoap* apropiada. Por otro lado, al ser PHP, el desarrollo de los servicios web ha sido completamente manual, programándolos en editores de texto tradicionales como el *Notepad*. De la misma manera, su depuración ha sido más complicada: al no existir herramienta de depuración, toda esta ha sido de forma manual con mensajes de información y error. No sin olvidarnos de que ha sido necesario aprender además el lenguaje PHP, pues ninguno de los componentes del grupo tenía conocimientos previos.

Por todo ello, el tiempo dedicado al aprendizaje de los servicios web en PHP ha sido considerablemente mayor que para los servicios web implementados con J2EE. Además, según lo comentado anteriormente, hemos de destacar que el framework de J2EE ha facilitado la implementación de los servicios web utilizados para Sakai, siendo una labor más complicada para los servicios web de Moodle.



## 4. INTEGRACIÓN DE PLATAFORMAS EN CAMPUS VIRTUALES

### 4.1 INTRODUCCIÓN

El uso de tecnologías de la información y la comunicación (TIC) ha sido del interés de la comunidad científica durante años (Guri-Rosemblit 05). En la actualidad el uso de las TIC en educación suele denominarse e-learning (Kaplan-Leison 09), y se ha convertido en un factor de éxito para universidades e instituciones educativas (Guri-Rosemblit 05, Leidner 95, Neville 05). Así, las universidades tradicionales están evolucionando total o parcialmente hacia universidades virtuales (PLS 04, Whittington 98). El término “universidad virtual” se utiliza para describir distintos tipos de universidades que ofrecen sus cursos en formato e-learning, con independencia de que la universidad en cuestión imparta docencia presencial o a distancia (PLS 04). Esta situación, ha promovido el auge de campus virtuales. En un informe de hace algunos años, Van Dusen y Gerald dieron la siguiente definición de campus virtual: “Los campus virtuales son una metáfora para la docencia electrónica, entornos de investigación y docencia creados por la convergencia de tecnologías relativamente nuevas, incluyendo, pero no restringidas a Internet, la World Wide Web, comunicación basada en computadores, videoconferencia, multimedia, trabajo colaborativo, vídeo bajo demanda, autoedición, tutores inteligentes y realidad virtual” (Van Dusen 97). En un estudio más reciente sobre modelos de universidades europeas llevado a cabo por la comisión europea (PLS 04), los campus virtuales se entienden, en un sentido amplio, como la integración las TIC en las estructuras educativas organizativas de las universidades. Este es también el caso de muchos informes y estudios llevados a cabo por distintas organizaciones americanas (Dewey 06, Epper 04, Green 05). En el proyecto AACV, y en este trabajo, los campus virtuales son considerados como aplicaciones software que las instituciones educativas superiores utilizan para dar soporte a sus actividades de investigación y docencia.

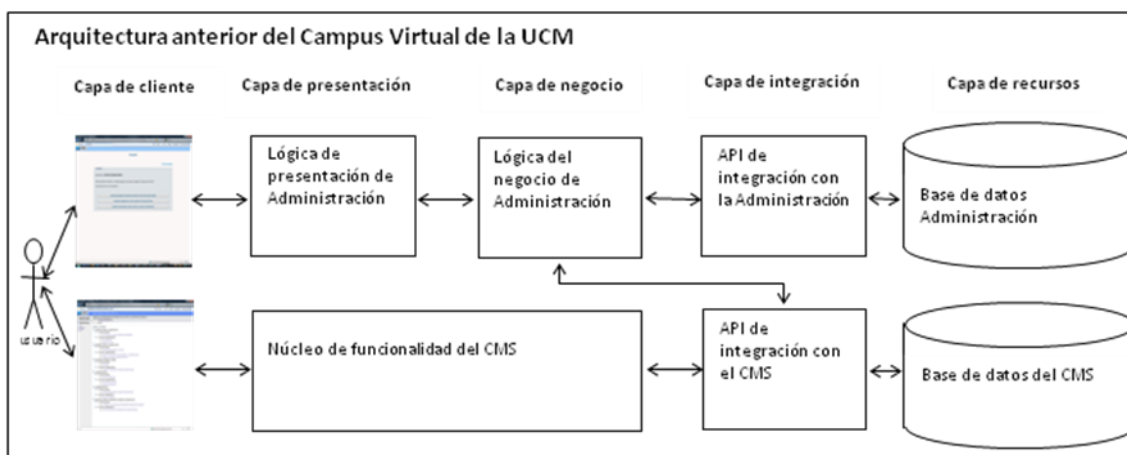
En este contexto, los campus virtuales suelen utilizar un CMS (Course Management System) para dar soporte a dichas actividades. La experiencia demuestra que los campus virtuales de nueva generación se están convirtiendo en aplicaciones web de dimensión industrial que tienen que afrontar el reto de integración de distintos CMS, de servicios de e-learning complementarios a éstos (por ejemplo, autoría de material educativo), y de servicios de gestión académica. En la actualidad, en el seno de las aplicaciones empresariales de dimensión industrial, existen diversos mecanismos de integración bien documentados (Hohpe 07). Estos mecanismos se basan en diferentes arquitecturas: multicapa (Alur 03), de integración (Hohpe 03), y orientada a servicios (Erl 05). La arquitectura multicapa define cinco capas en el diseño de una aplicación web (clientes, presentación, negocio, integración y recursos), y ha tenido bastante repercusión en el diseño de aplicaciones web de dimensión industrial (Fowler 03). La integración de aplicaciones de dimensión industrial incluye diversos patrones bien documentados (de estilos de integración, de canal, de construcción de mensajes, de direccionamiento, de transferencia, de extremo y de gestión de sistemas) extraídos de la experiencia en la integración de distintas plataformas (Hohpe 03). Por último, la arquitectura orientada a servicios presenta un modelo en el que la funcionalidad se descompone en pequeñas unidades denominadas servicios, que pueden estar distribuidos en una red y pueden ser combinados juntos y reutilizados para crear aplicaciones de empresariales (Newcomer 05). Estos servicios se comunican entre si pasando datos de un servicio a otro, o coordinando una actividad entre dos o más servicios (Erl 05)

El proyecto AACV y este proyecto de sistemas informáticos apuestan por una arquitectura orientada a servicios para la integración de distintas plataformas en campus virtuales. Las próximas secciones describen la arquitectura inicial del Campus Virtual de la universidad

complutense de Madrid, la arquitectura actual y la propuesta de arquitectura que hace el proyecto AACV y que desarrolla este proyecto de sistemas informáticos.

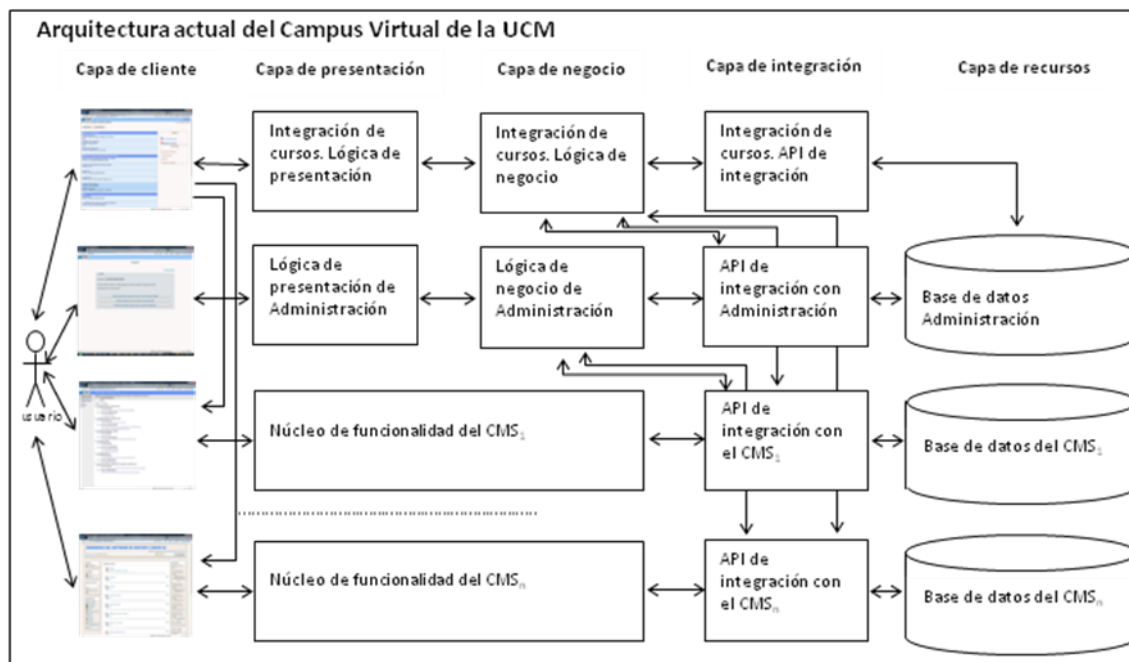
## 4.2 ARQUITECTURAS PRECEDENTES

Considerando las diferentes opciones sobre las cuales se podrían basar el campus virtual (CMS, LMS, LCMS), el Campus Virtual de la UCM fue basado desde su concepción en el año académico 2008-2009 en un CMS concreto (WebCT 4.1) complementado con una aplicación web administrativa desarrollada por la Unidad de Apoyo Técnico y Docente al Campus Virtual (UATD). El CMS es responsable de las funcionalidades centrales e-learning (por ejemplo, publicación de materiales para el aprendizaje), mientras la aplicación web es responsable de las funcionalidades administrativas complementarias (por ejemplo, cambio de contraseña). La Figura 4.1 (Navarro 10) representa este campus virtual a partir de una arquitectura multicapa (Alur 03). Nótese que, de hecho, durante estos años, el campus virtual de la UCM fue una aplicación desarrollada en PHP sin seguir los patrones de arquitectura multicapa. Sin embargo, por cuestiones de homogeneidad con el resto de arquitecturas descritas, se muestra en la Figura 4.1 una arquitectura multicapa.



**FIGURA 4.1 ARQUITECTURA ANTERIOR DEL CAMPUS VIRTUAL DE LA UCM**

Las presiones derivadas del coste económico de las licencias comerciales del CMS utilizado por el Campus Virtual de la UCM, así como las presiones de los usuarios, llevó al uso paralelo de CMSs adicionales en el Campus Virtual UCM. Así, durante varios cursos dos CMSs (Moodle 1.9.2 y Sakai 2.4.0) fueron probados por los usuarios del Campus Virtual de la UCM. En este curso académico 2009-2010 un adicional CMS (Moodle) adicional ha sido desplegado totalmente en el Campus Virtual de la UCM. Además, para facilitar la interacción de los usuarios con más de un CMS, se ha desarrollado una infraestructura de integración, que representa en una página común los cursos que un mismo usuario puede tener en diferentes CMSs. La Figura 4.2 (Navarro 10) representa esta situación. En este caso, se desarrolló una arquitectura multicapas implementada en J2EE.



**FIGURA 4.2 ARQUITECTURA ACTUAL DEL CAMPUS VIRTUAL DE LA UCM**

Cada CMS contiene los diferentes cursos que se hayan virtualizado en el mismo. Los cursos existirán únicamente en la plataforma que se haya elegido; por este motivo los usuarios deberán acceder a las distintas plataformas.

La elección de CMS donde alojar un curso es una labor propia del usuario que virtualice el curso, por este motivo dicha elección será el resultado de la valoración de las funcionalidades y ventajas respecto al sistema de enseñanza del usuario.

El principal punto negativo de esta arquitectura se refiere visión del usuario, que resulta completamente diferente en cada plataforma, tanto visualmente como las funcionalidades disponibles.

### 4.3 ARQUITECTURA PROPUESTA

En esta sección se va a describir la nueva arquitectura para la nueva generación de campus virtuales propuesta en el proyecto AACV (Navarro 10) e implementada en este proyecto de Sistemas Informáticos. Se pretende proporcionar una arquitectura capaz de hacer a los usuarios independientes, tanto como sea posible, de los CMSs concretos que conformen el campus virtual, tomando como referencia el Campus Virtual UCM.

El objetivo es la propuesta de una arquitectura para campus virtuales que sea independiente del CMS subyacente, modificable en cuanto al tratamiento de las necesidades de los usuarios, en constante evolución y accesible para los mismos.

La arquitectura del actual campus virtual de la UCM, a pesar de disponer una página de integración, sus usuarios deben interactuar con los CMSs específico, por lo tanto, la dependencia de un CMS concreto ha sido duplicada.

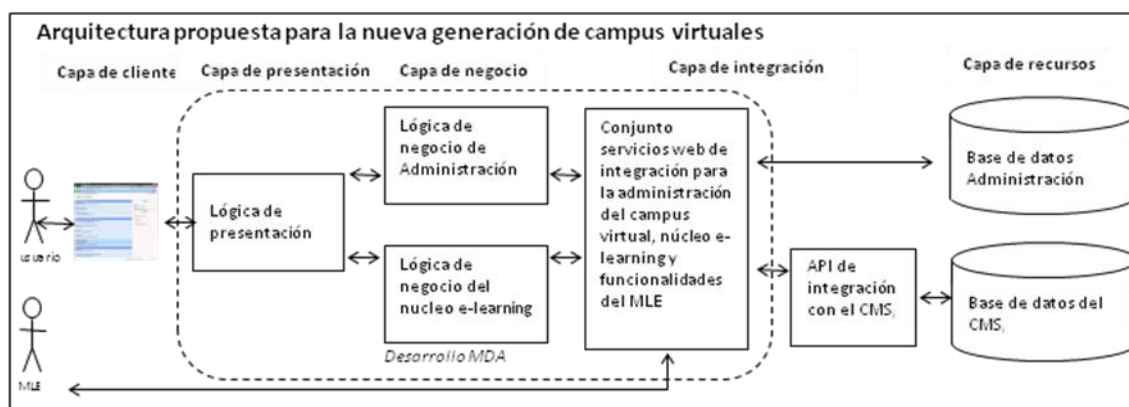
La nueva generación de campus serán concebidos como campus virtuales de *marca blanca* debido a que lo que se busca es la independencia de un CMS concreto (Navarro 10). Para esto, los campus virtuales de marca blanca pueden ser definidos como aquellos que son independientes de cualquier CMS subyacente. El objetivo de independencia es el intercambio

transparente entre CMSs, siempre que las herramientas de migración para los cursos estén disponibles.

El núcleo del campus virtual de marca blanca es el conjunto de interfaces de servicios web, implementados específicamente para un CMS concreto utilizado en el campus virtual. Estos servicios web pueden ser utilizados en una mayor infraestructura para la gestión universitaria. La Figura 4.3 (Navarro 10) muestra la arquitectura de este campus virtual. Un campus virtual solo puede basarse en un único CMS, o puede necesitar más de un CMS para hacer frente a las principales funcionalidades e-learning que necesitan los usuarios.

Aunque no haya sido utilizada en este proyecto de Sistemas Informáticos, el proyecto AACV basa promueve el uso de Arquitectura Dirigida por Modelos (Model Driven Architecture, MDA) (Navarro 08a, Navarro 08b) en el desarrollo de esta nueva arquitectura. El objetivo de MDA es el desarrollo software basado en la presencia de modelos de diferentes niveles de abstracción. La evolución de modelos abstractos a modelos concretos, y finalmente a código, se realiza a través de lenguajes de transformación (OMG 01).

Debido a la heterogeneidad de los usuarios (estudiantes, profesores e investigadores) y a la rápida evolución de las tecnologías e-learning, las técnicas de desarrollo software utilizadas debían ser capaces de hacer frente a cambios en los requisitos software que pudiesen aparecer. Por lo tanto, el desarrollo de funcionalidades en el campus virtual de marca blanca, así como el desarrollo de los servicios web, pueden beneficiarse del uso de las técnicas MDA. La figura 4.3 muestra el uso del desarrollo MDA.



**FIGURA 4.3 ARQUITECTURA PROPUESTA PARA LA NUEVA GENERACIÓN DE CAMPUS VIRTUALES**

La siguiente sección describe el diseño de la solución para una arquitectura de integración basada en la Figura 4.3.



## 5. DISEÑO DE LA SOLUCIÓN

### 5.1 INTRODUCCIÓN

Este apartado de la memoria tiene como objetivo la definición de la arquitectura del sistema y del entorno tecnológico en la que se despliega, junto con la especificación detallada de los componentes de la aplicación y la distinta funcionalidad que el usuario puede llevar a cabo manejando el nuevo campus virtual.

Esta descripción incluye la identificación de las funcionalidades que se ofrecen representadas mediante los diagramas de casos de uso así como los esquemas de alto nivel de cada uno de ellos (diagramas de actividad); la arquitectura del sistema, junto a las clases, interfaces y la relación entre las mismas y la interacción de los objetos en las funcionalidades más complejas; el paso de mensajes que se requiere para llevar a cabo la funcionalidad que se representa (diagramas de secuencia); la distribución de componentes que forman el sistema y las dependencias entre estos, y la representación del hardware utilizado en la implementación del sistema y las relaciones entre sus componentes, información que se refleja en el diagrama de despliegue.

Así mismo, se muestra el diagrama de entidad-relación de la base de datos Campus Virtual, indispensable para asegurar la correcta administración del campus virtual que se ha implementado.

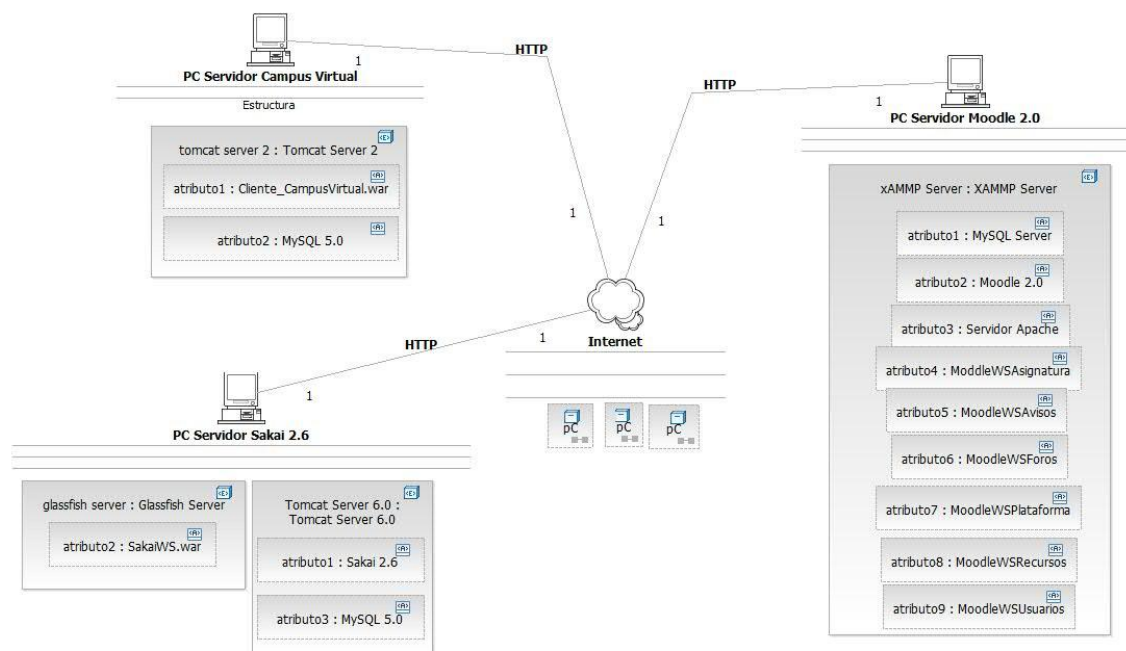
### 5.2 ARQUITECTURA

Para explicar el diseño de la solución tenemos que hablar de dos aplicaciones diferentes. La primera es con la que interactúa el usuario, es la denominada Aplicación Campus Virtual. La otra es el servicio web que hace de intermediario entre el Campus Virtual y la plataforma en cuestión. Como en este proyecto hemos desarrollado servicios web para las plataformas Sakai 2.6 y Moodle 2.0 tenemos dos aplicaciones diferentes que interactúan con cada una de las plataformas.

Las tres aplicaciones están diseñadas mediante una arquitectura en capas. Para el Campus Virtual tenemos tres capas: la capa de presentación, la capa de lógica de negocio y la capa de integración. Sin embargo en los servicios web simplemente tenemos las capas de lógica de negocio y la capa de integración.

#### 5.2.1 DIAGRAMA DE DESPLIEGUE

En la siguiente figura podemos observar la arquitectura general que se ha propuesto para este desarrollo del proyecto.



**FIGURA 5.1 DIAGRAMA DE DESPLIEGUE**

Cada una de las plataformas utilizadas se puede instalar en máquinas diferentes, bajo los servidores que utilizan: Sakai 2.6 está desplegado bajo Tomcat 6.0, y Moodle 2.0 sobre el servidor Apache de XAMPP. A su vez, cada plataforma utiliza una base de datos diferente, en ambos casos se trata de MySQL. El servidor XAMPP de Moodle ya la incorpora, pero en el caso de Sakai es necesario instalar y configurar MYSQL 5.0 para que Sakai pueda acceder a ella.

Los servicios web de Moodle se encuentran bajo el servidor XAMPP, pues necesitan acceder directamente a las APIs de la plataforma. Sin embargo, los servicios web de Sakai acceden a un servicio web de la plataforma que también se ha adaptado, por ello, estos servicios web (SakaiWS.war) pueden desplegarse bajo otro servidor que los despliegue, por ejemplo, Glassfish Server.

Por otro lado, se encuentra el cliente que consume estos servicios web, es decir, el Campus Virtual de marca blanca Cliente\_CampusVirtual.war. Al tratarse de una aplicación web J2EE que contiene servlets y las referencias a los servicios web, puede desplegarse bajo el servidor de Tomcat (aunque en este proyecto ha sido desplegado sobre Glassfish). Además, este cliente accede a la base de datos del Campus Virtual que por comodidad se ha implementado también bajo MySQL 5.0.

## 5.3 DISEÑO CAMPUS VIRTUAL

### 5.3.1 INTRODUCCIÓN

Como se ha mencionado anteriormente, la aplicación del Campus Virtual es la que interactúa con el usuario, y la que se comunica con los servicios web de cada plataforma. Este campus virtual de marca blanca utiliza los patrones de MVC, Factoría Abstracta y Singleton (Gamma 06).

El patrón MVC (Model-View-Controller) define la organización independiente del Modelo (los datos de una aplicación), la Vista (la interfaz de usuario) y el Controlador (la lógica de control). Para su implementación se ha optado por utilizar el framework de Struts 1.3.8, disponible en <http://struts.apache.org/>. Es una herramienta de desarrollo para aplicaciones Web bajo este patrón para la plataforma J2EE. Su elección se debe a que permite reducir el tiempo de desarrollo, por su carácter de software libre, su solidez, y por ser uno de los frameworks más utilizados actualmente en el mundo laboral.

El patrón Singleton asegura que exista una única instancia de una clase, convirtiendo a este patrón en indispensable para la arquitectura propuesta, debido a la necesidad de ocultar el constructor de la clase, para que no se puedan crear instancias y la necesita de una única referencia a la instancia.

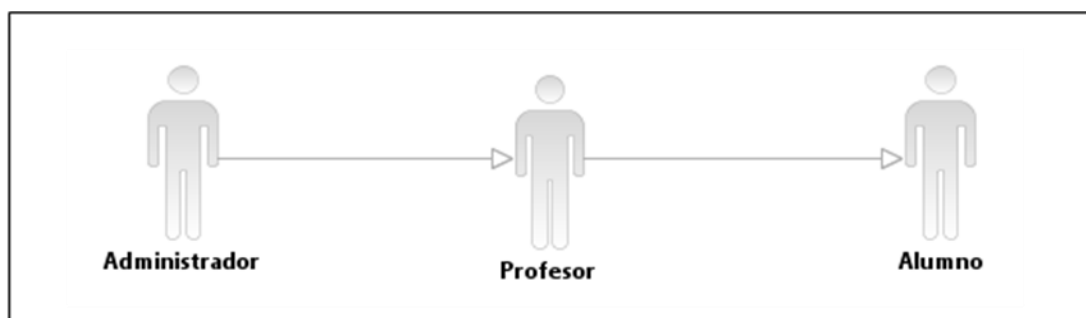
El patrón Factoría Abstracta se convierte en una necesidad en la arquitectura ya que solo debe existir una única interfaz que proporcione toda la funcionalidad del campus virtual. Este resultado puede obtenerse a través del uso de este patrón, que permite la creación de tantos objetos de este tipo como sean necesarios y en el caso del campus virtual de marca blanca, de tantos CMS como disponga, manteniendo una funcionalidad conjunta y equivalente a ultimar para cada CMS.

La arquitectura de la aplicación se ha diseñado mediante la organización en capas. Así, en esta aplicación nos encontramos con las capas de Presentación, de Lógica de Negocio y de Integración, que más adelante se detallan.

### 5.3.2 CASOS DE USO

Los casos se han dividido en módulos, cuyo dominio está bien definido, siendo estos módulos Usuarios, Asignaturas, Recursos, Foros, Anuncios y Plataforma. Cada uno de estos módulos interactuará con los diferentes perfiles de usuarios, dependiendo del nivel de permisos otorgado a cada uno de estos perfiles.

La herencia de funcionalidades entre los usuarios se representa en la Figura 5.1



**FIGURA 5.2 HERENCIA DE USUARIOS**

No obstante, la descripción completa de los casos de uso se encuentra en el Apéndice A.

A continuación, se describen los casos de uso de cada uno de los módulos definidos, junto a sus correspondientes diagramas de actividad, que representan el flujo paso a paso de cada uno de ellos.

## MÓDULO USUARIOS

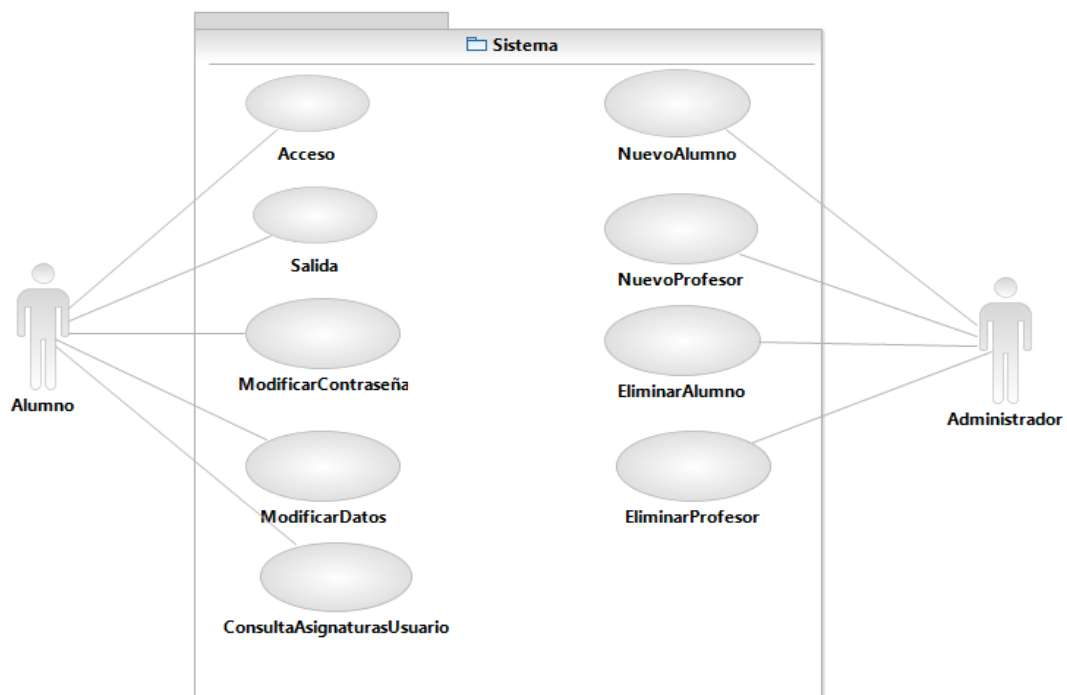
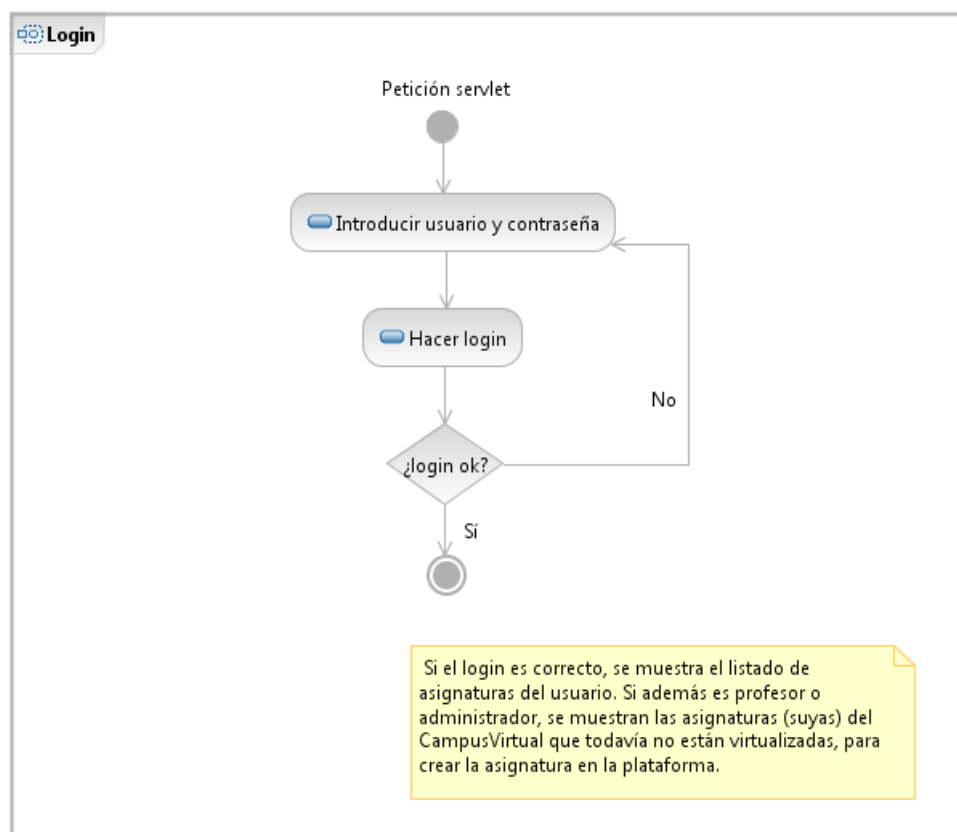


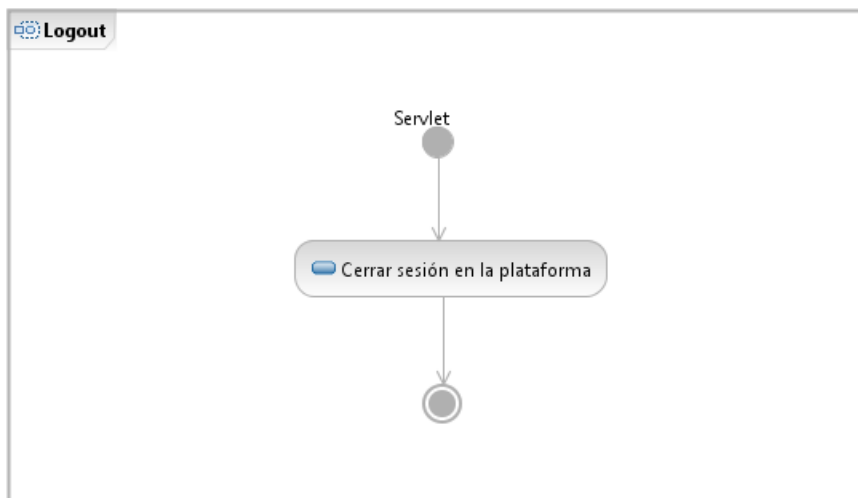
FIGURA 5.3 CASOS DE USO - USUARIOS

Caso de uso "Acceso": Gestiona el acceso al Campus Virtual, mediante el nombre de usuario y la contraseña.



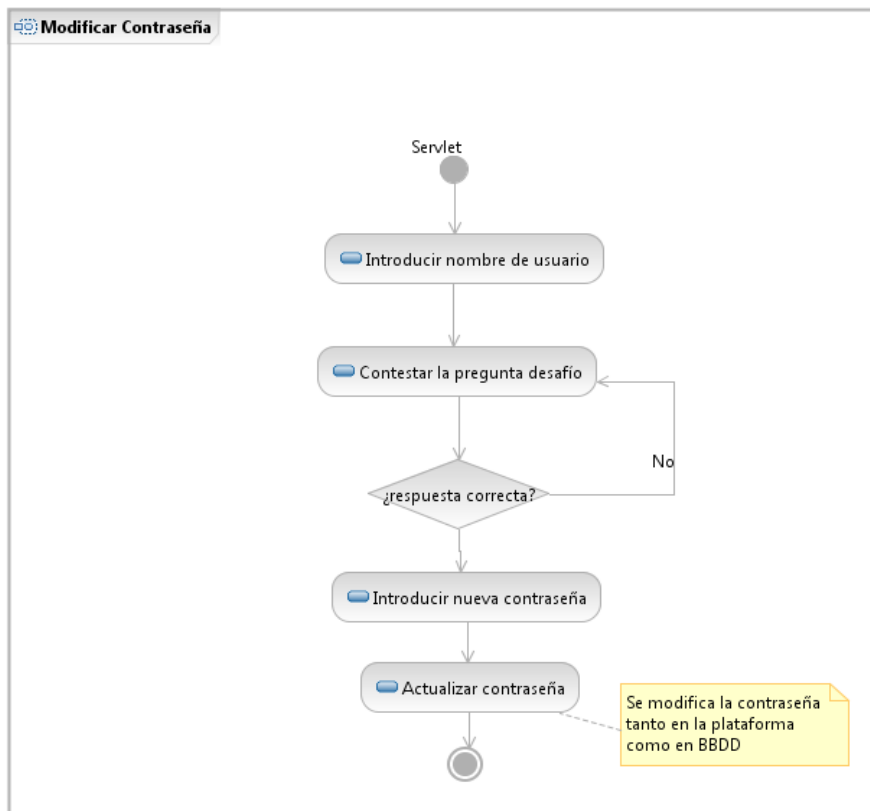
**FIGURA 5.4 DIAGRAMA DE ACTIVIDAD - ACCESO**

Caso de uso “Salida”: el usuario realiza la desconexión del Campus Virtual



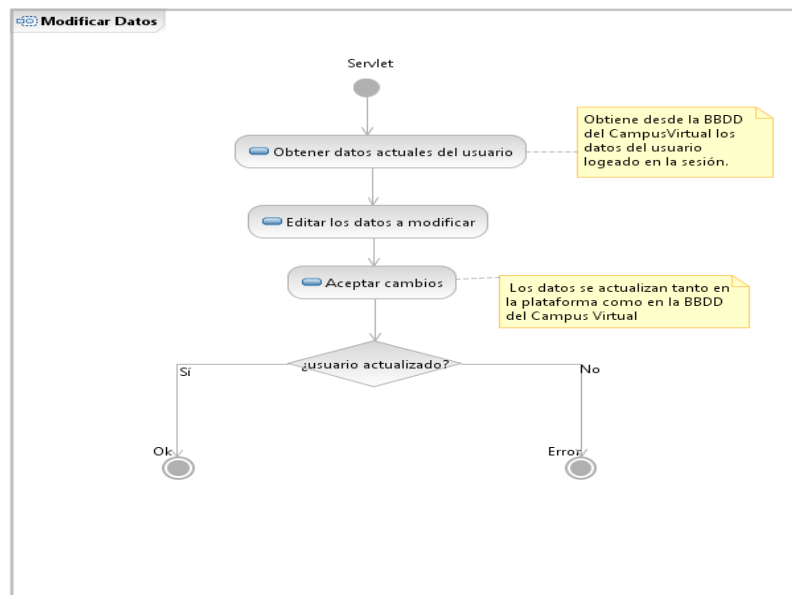
**FIGURA 5.5 DIAGRAMA DE ACTIVIDAD - SALIDA**

Caso de uso “Modificar Contraseña”: cuando el usuario olvida su contraseña de acceso, puede solicitar una nueva clave. Para ello, deberá responder correctamente a la pregunta desafío y en tal caso, modificar su contraseña.



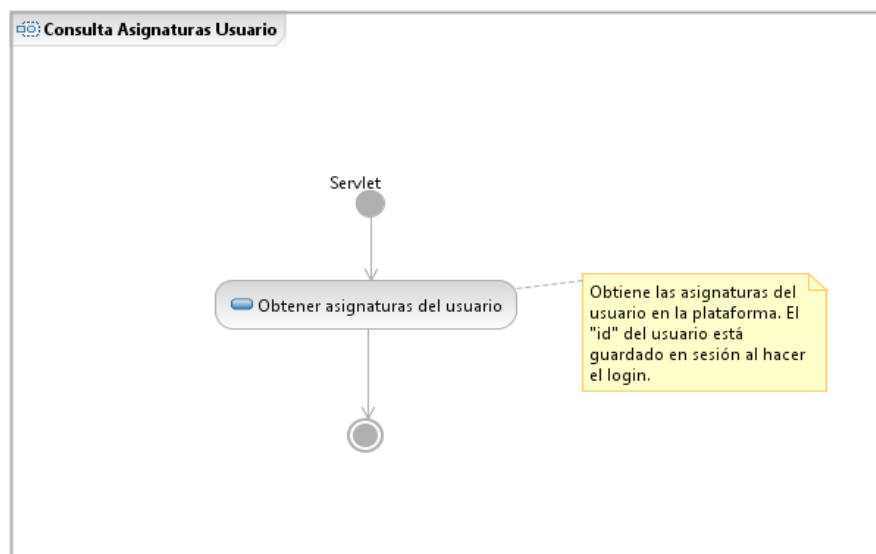
**FIGURA 5.6 DIAGRAMA DE ACTIVIDAD – MODIFICAR CONTRASEÑA**

Caso de uso “Modificar Datos”: el usuario puede modificar sus datos personales, como son el nombre y apellidos, el e-mail, la contraseña de acceso y la pregunta y respuesta desafío.



**FIGURA 5.7 DIAGRAMA DE ACTIVIDAD – MODIFICAR DATOS**

Caso de uso “Consulta Asignaturas Usuario”: obtiene las asignaturas virtualizadas del usuario que está logueado en el Campus Virtual.



**FIGURA 5.8 DIAGRAMA DE ACTIVIDAD – CONSULTA ASIGNATURAS USUARIO**

Caso de uso “Nuevo Alumno”: un profesor o administrador puede dar de alta a un alumno en el Campus Virtual. Para ello, debe rellenar los datos del alumno.

Caso de uso “Nuevo Profesor”: un profesor o administrador puede dar de alta a otro profesor en el Campus Virtual. Para ello, debe rellenar los datos del profesor.

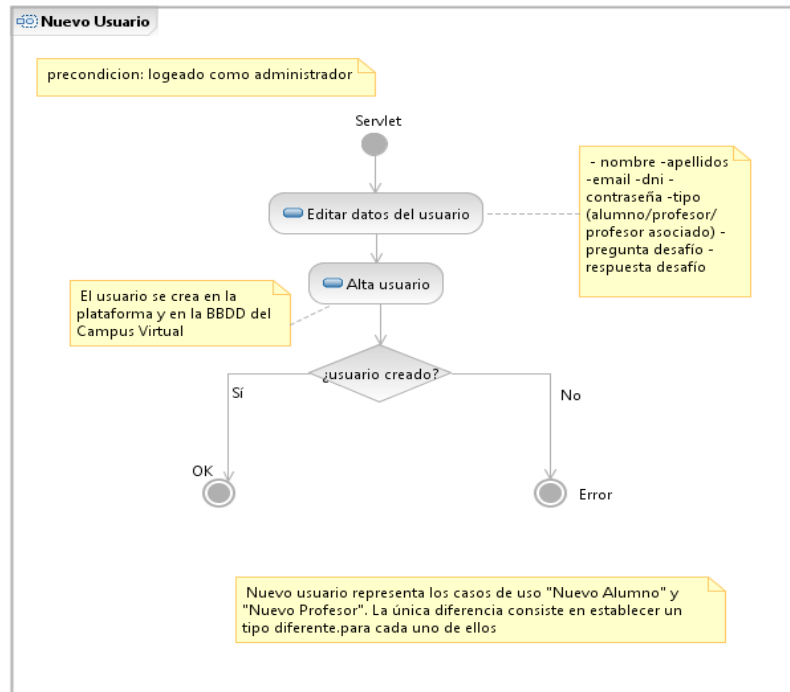


FIGURA 5.9 DIAGRAMA DE ACTIVIDAD – NUEVO USUARIO

Caso de uso “Eliminar Alumno”: un profesor o administrador puede dar de baja a un alumno en el Campus Virtual. El alumno se elimina tanto de la plataforma como de la base de datos del Campus Virtual.

Caso de uso “Eliminar Profesor”: un profesor o administrador puede dar de baja a un profesor en el Campus Virtual. El profesor se elimina tanto de la plataforma como de la base de datos del Campus Virtual.

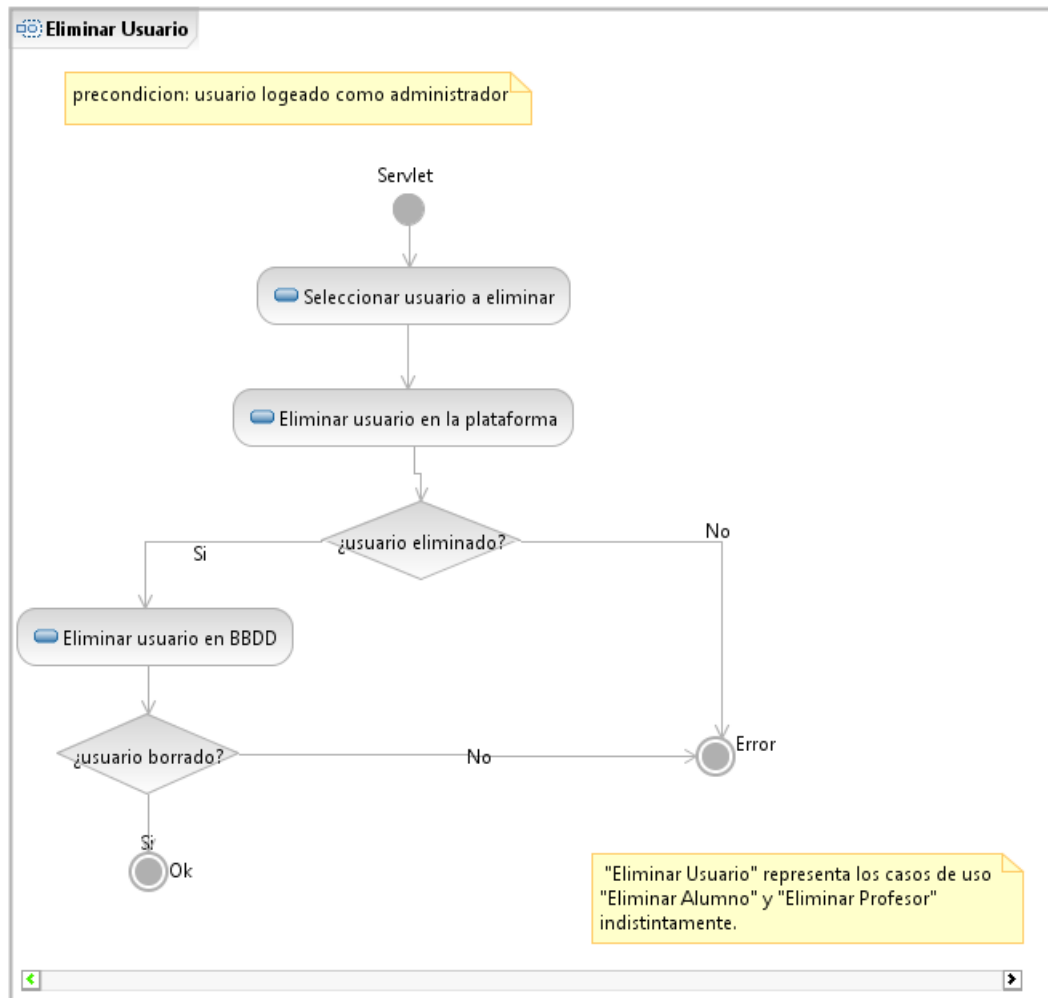


FIGURA 5.10 DIAGRAMA DE ACTIVIDAD – ELIMINAR USUARIO



## MÓDULO ASIGNATURAS



FIGURA 5.11 CASOS DE USO - ASIGNATURAS

Caso de uso “Nueva Asignatura”: Para virtualizar una asignatura, el profesor debe elegir entre las suyas que aún no estén virtualizadas. Tras rellenar los datos de la misma, la asignatura se crea en la plataforma con las herramientas necesarias para su uso, como son por ejemplo la creación automática de un foro.

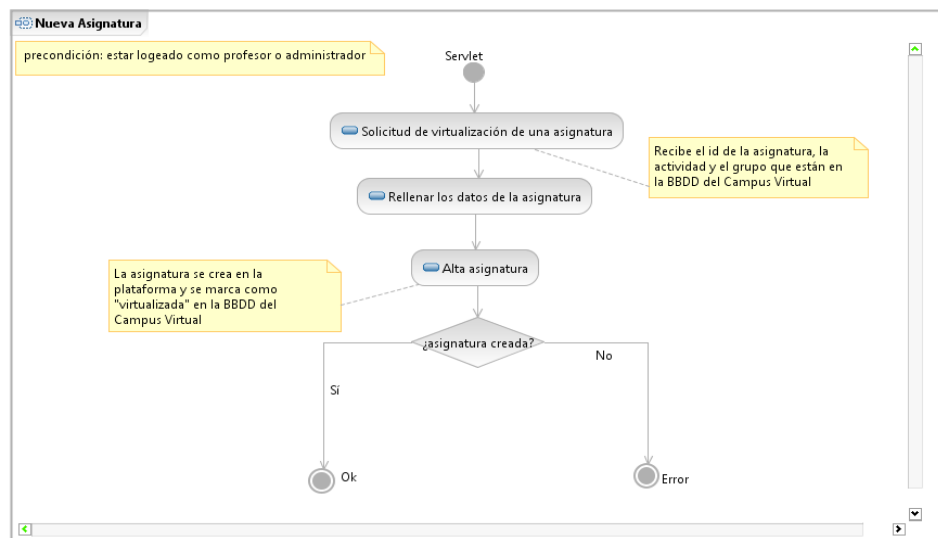


FIGURA 5.12 DIAGRAMA DE ACTIVIDAD – NUEVA ASIGNATURA

Caso de uso “Alta Alumno Asignatura”: Añade a un usuario a una asignatura como alumno.

Caso de uso “Alta Profesor Asignatura”: Añade a un usuario a una asignatura como profesor.

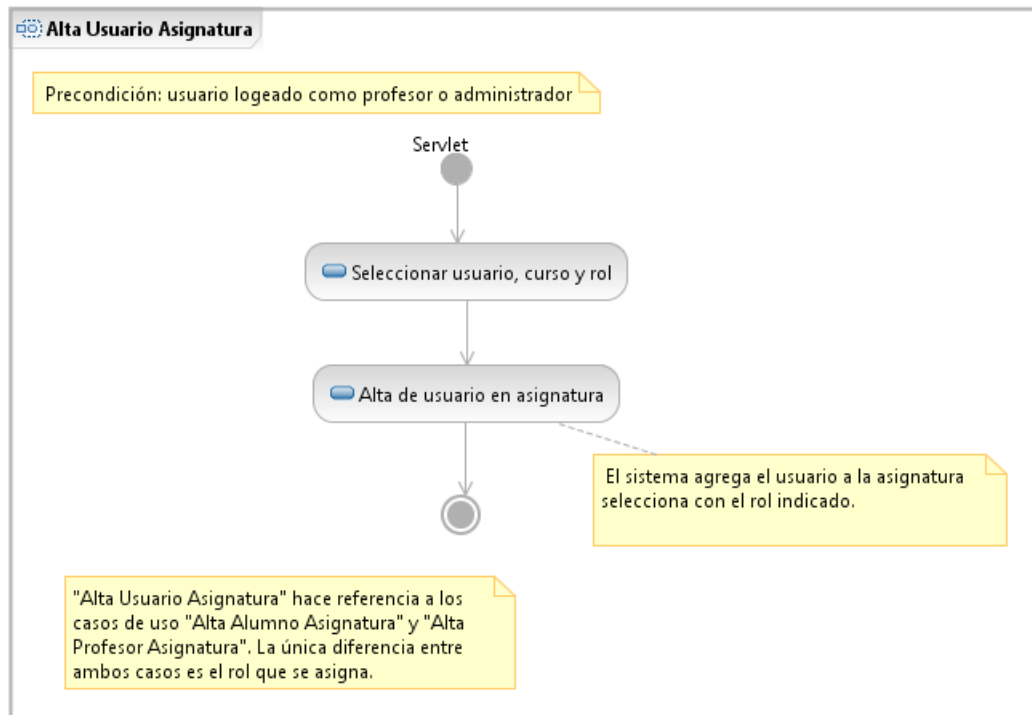


FIGURA 5.13 DIAGRAMA DE ACTIVIDAD – ALTA USUARIO ASIGNATURA

Caso de uso “Baja Alumno Asignatura”: Elimina a un alumno de una asignatura.

Caso de uso “Baja Profesor Asignatura”: Elimina a un profesor de una asignatura.

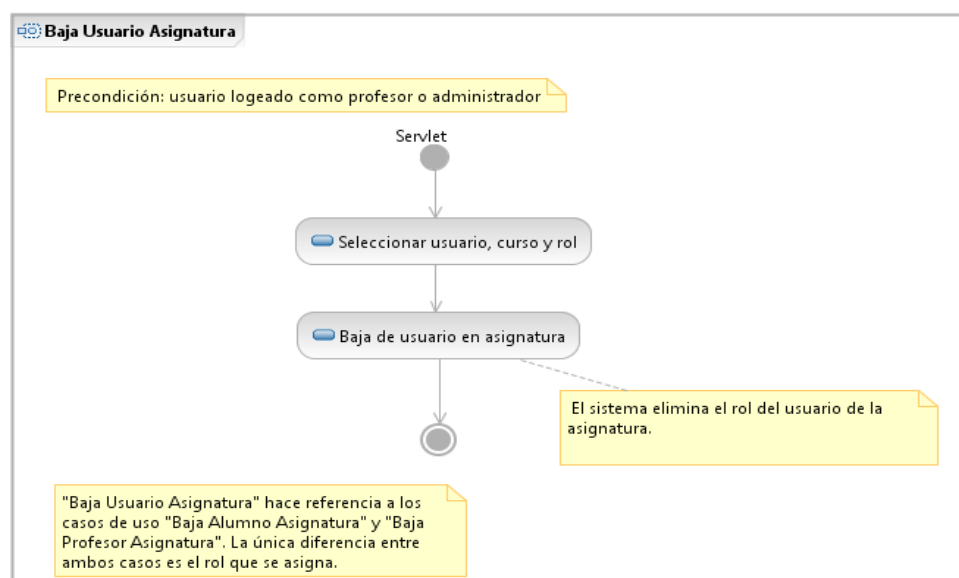


FIGURA 5.14 DIAGRAMA DE ACTIVIDAD – BAJA USUARIO ASIGNATURA

## MÓDULO RECURSOS

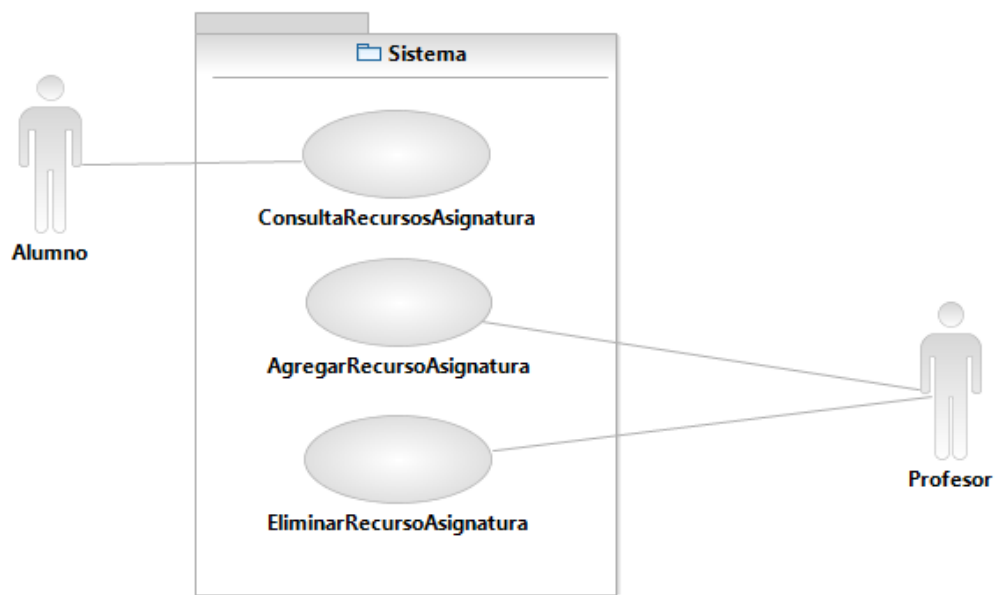


FIGURA 5.15 CASOS DE USO - RECURSOS

Caso de uso “Consulta Recursos Asignatura”: obtiene la lista de recursos de la asignatura seleccionada.

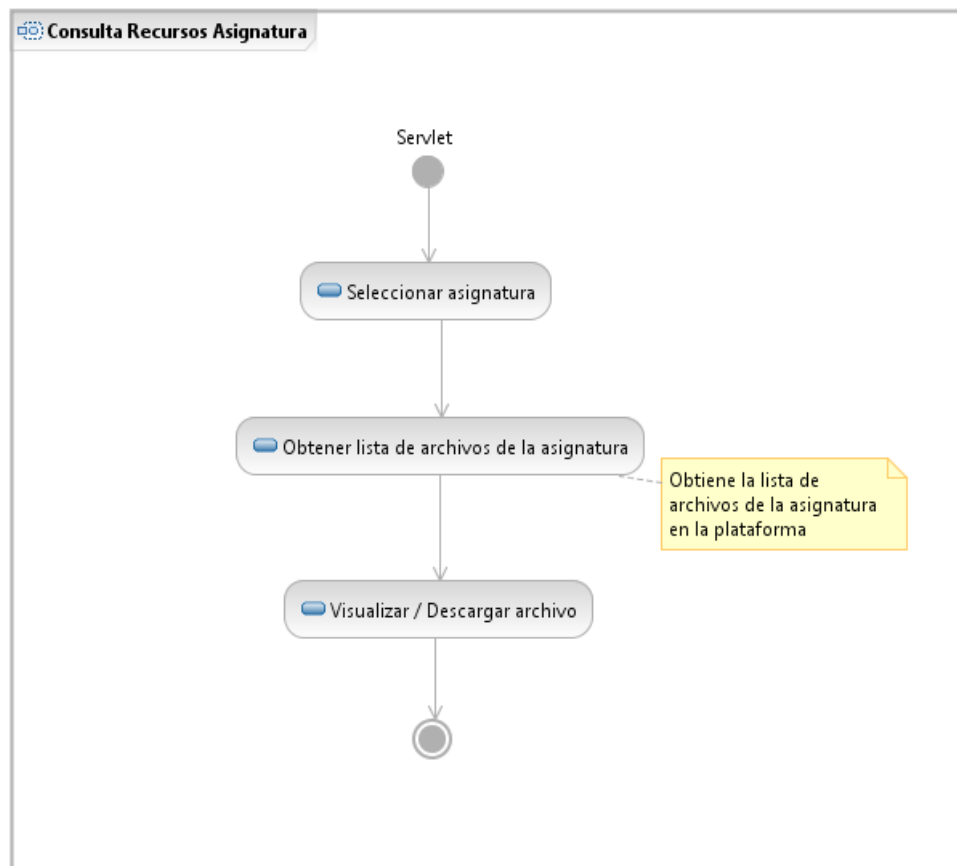


FIGURA 5.16 DIAGRAMA DE ACTIVIDAD – CONSULTA RECURSOS ASIGNATURA

Caso de uso “Agregar Recurso Asignatura”: añade un archivo en la asignatura seleccionada.

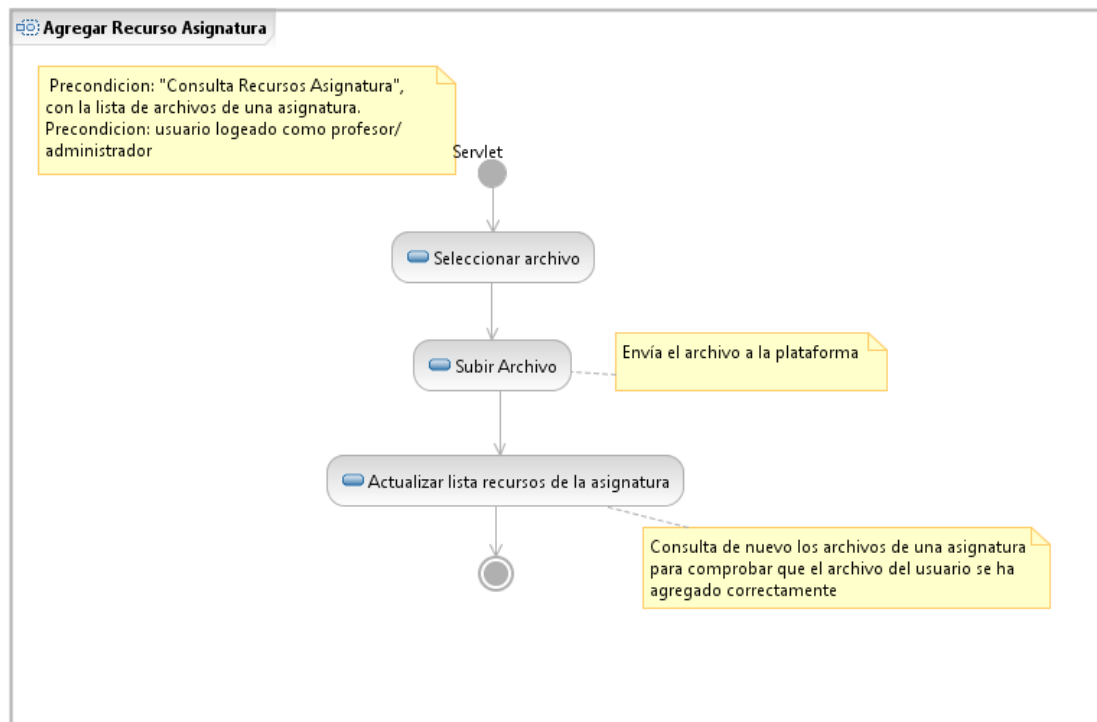


FIGURA 5.17 DIAGRAMA DE ACTIVIDAD – AGREGAR RECURSO ASIGNATURA

Caso de uso “Eliminar Recurso Asignatura”: borra un recurso de la asignatura indicada.

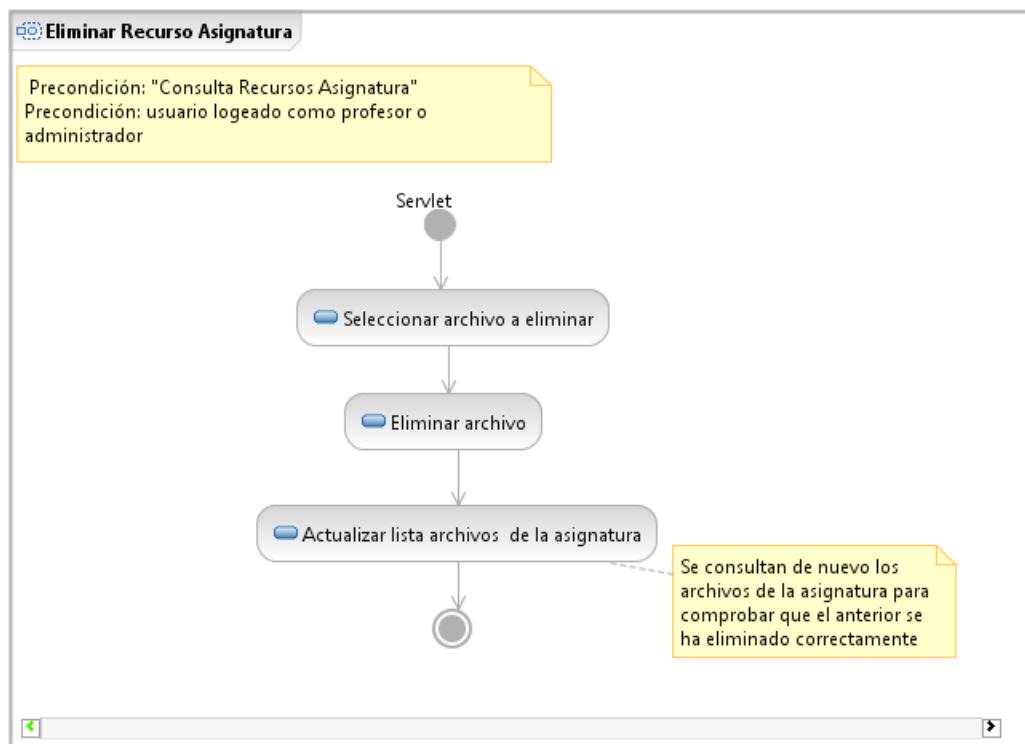


FIGURA 5.18 DIAGRAMA DE ACTIVIDAD – ELIMINAR RECURSO ASIGNATURA

## MÓDULO FOROS

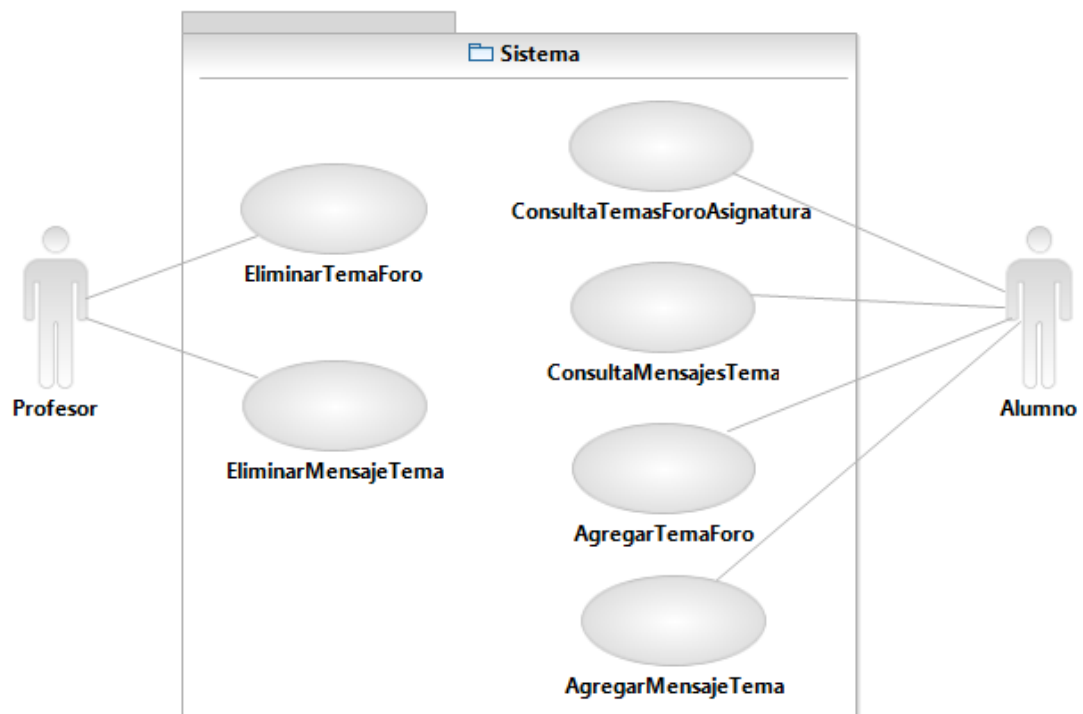


FIGURA 5.19 CASOS DE USO - FOROS

Caso de uso “Consulta Temas Foro Asignatura”: obtiene la lista de temas del foro de la asignatura seleccionada.

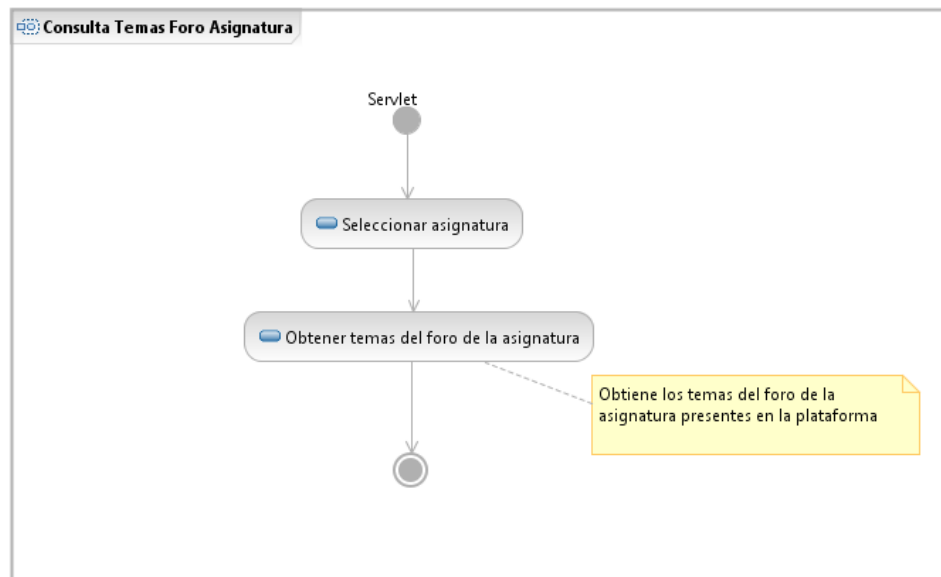


FIGURA 5.20 DIAGRAMA DE ACTIVIDAD – CONSUTLA TEMAS FORO ASIGNATURA

Caso de uso “Consulta Mensajes Tema”: obtiene la lista de mensajes del tema seleccionado del foro de la asignatura.

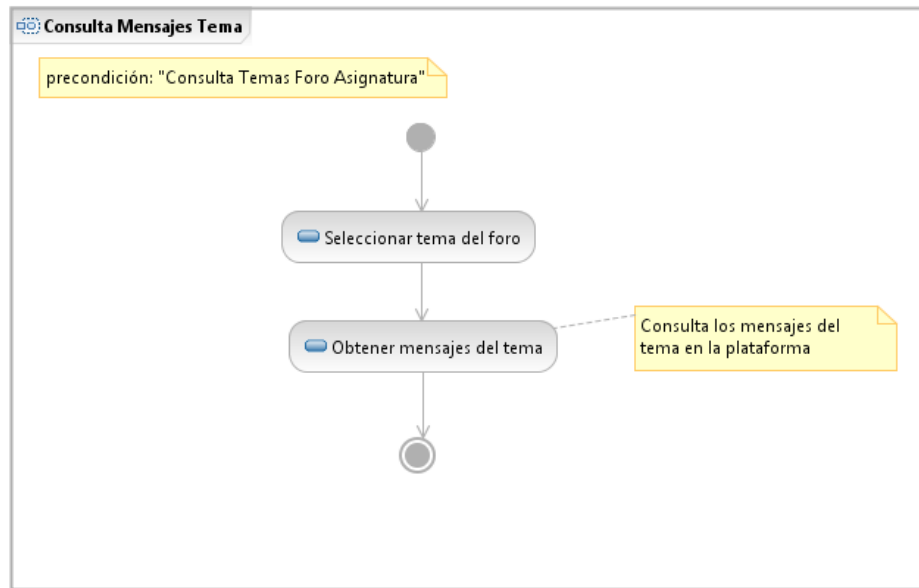


FIGURA 5.21 DIAGRAMA DE ACTIVIDAD – CONSULTA MENSAJES TEMA

Caso de uso “Agregar Tema Foro”: Añade un nuevo tema al foro de la asignatura. Para ello, se debe indicar el título de dicho tema.

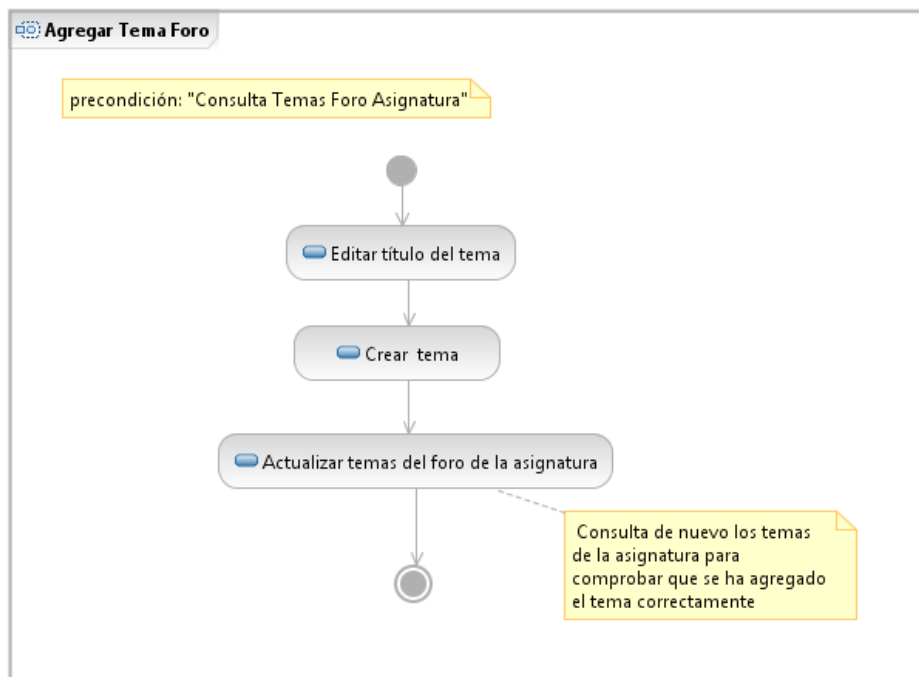
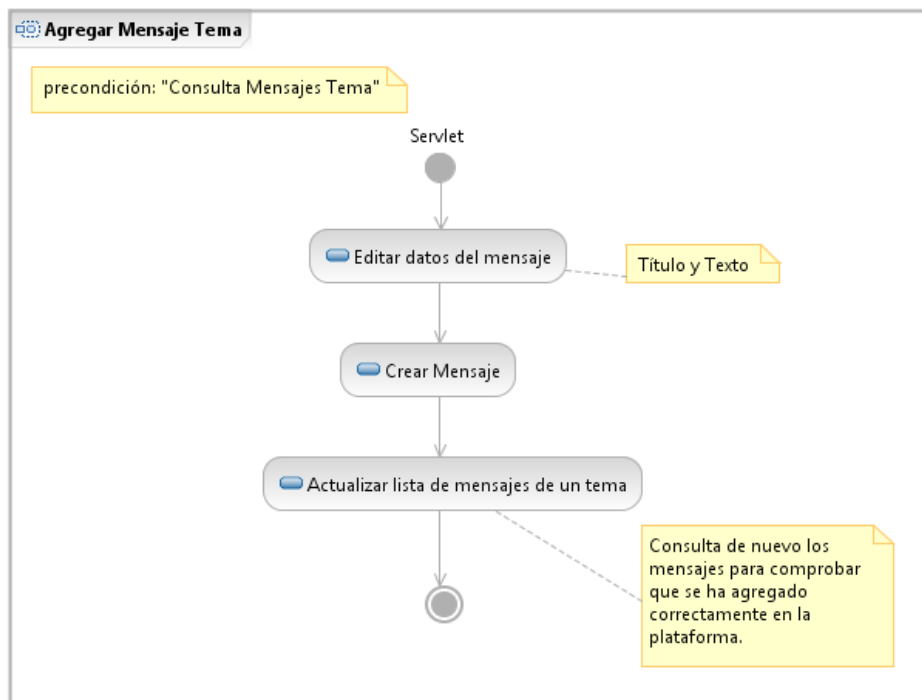


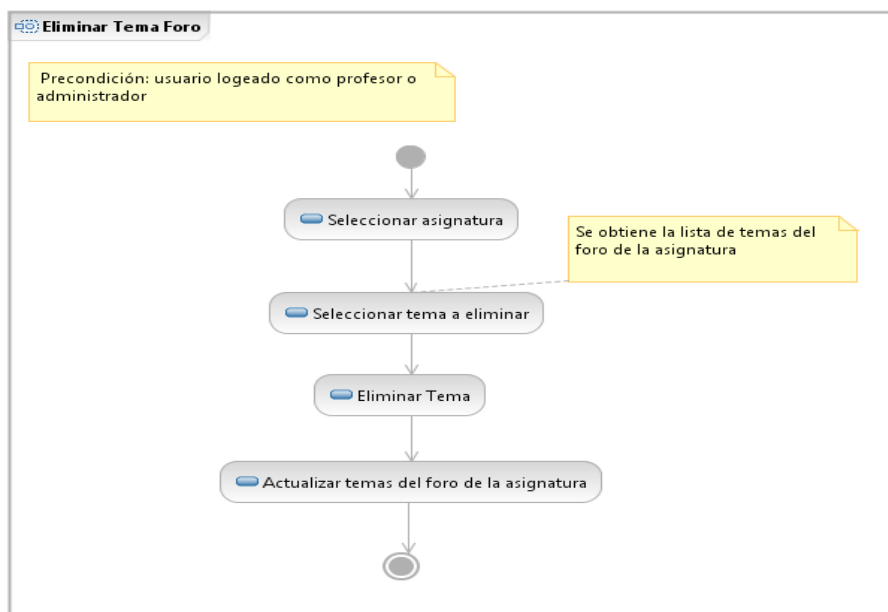
FIGURA 5.22 DIAGRAMA DE ACTIVIDAD – AGREGAR TEMA FORO

Caso de uso “Agregar Mensaje Tema”: Añade un nuevo mensaje al tema seleccionado de la asignatura. Para ello se debe editar el título y el texto del mensaje.



**FIGURA 5.23 DIAGRAMA DE ACTIVIDAD – AGERGAR MENSAJES TEMA**

Caso de uso “Eliminar Tema Foro”: un profesor puede seleccionar la opción de eliminar un tema del foro de una asignatura, eliminando así automáticamente los mensajes que pertenezcan al mismo.



**FIGURA 5.24 DIAGRAMA DE ACTIVIDAD – ELIMINAR TEMA FORO**

Caso de uso “Eliminar Mensaje Foro”: un profesor puede eliminar un mensaje de un tema del foro de la asignatura.

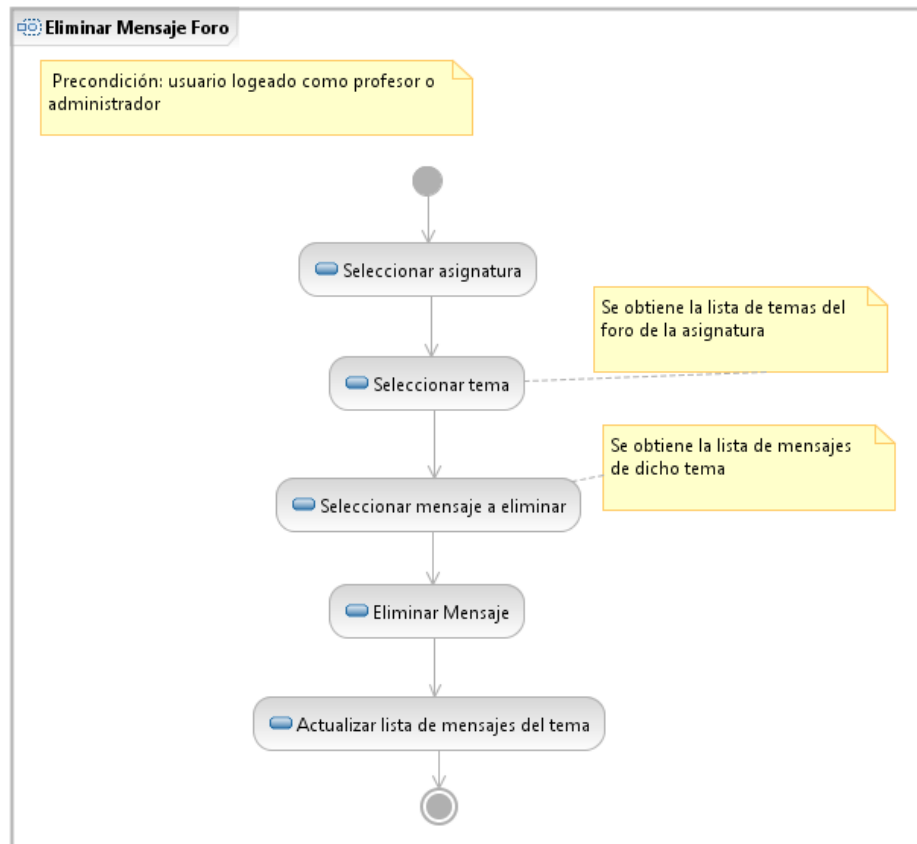


FIGURA 5.25 DIAGRAMA DE ACTIVIDAD – ELIMINAR MENSAJE FORO

## MÓDULO ANUNCIOS

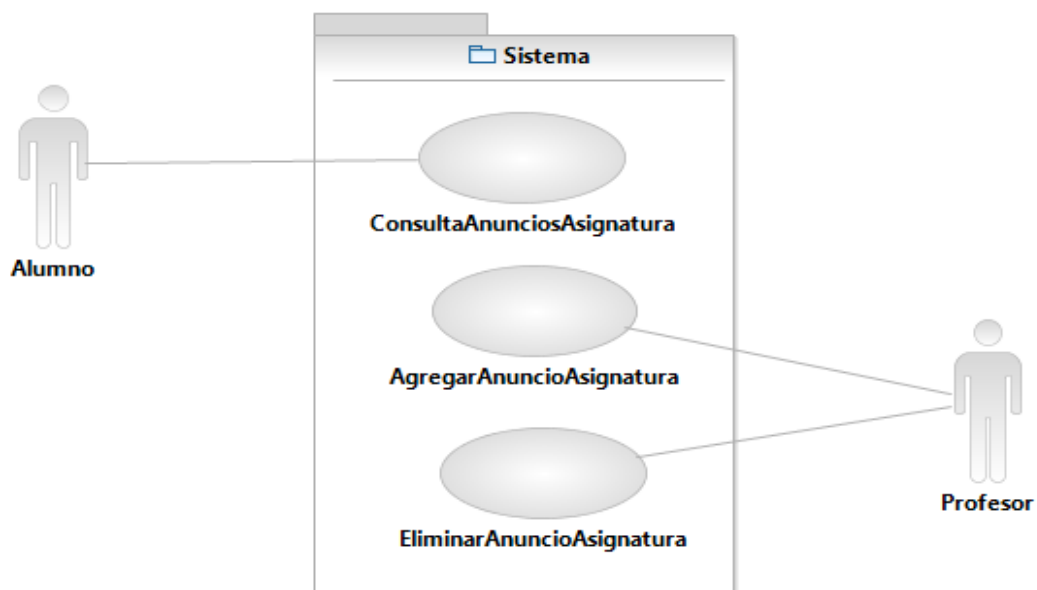
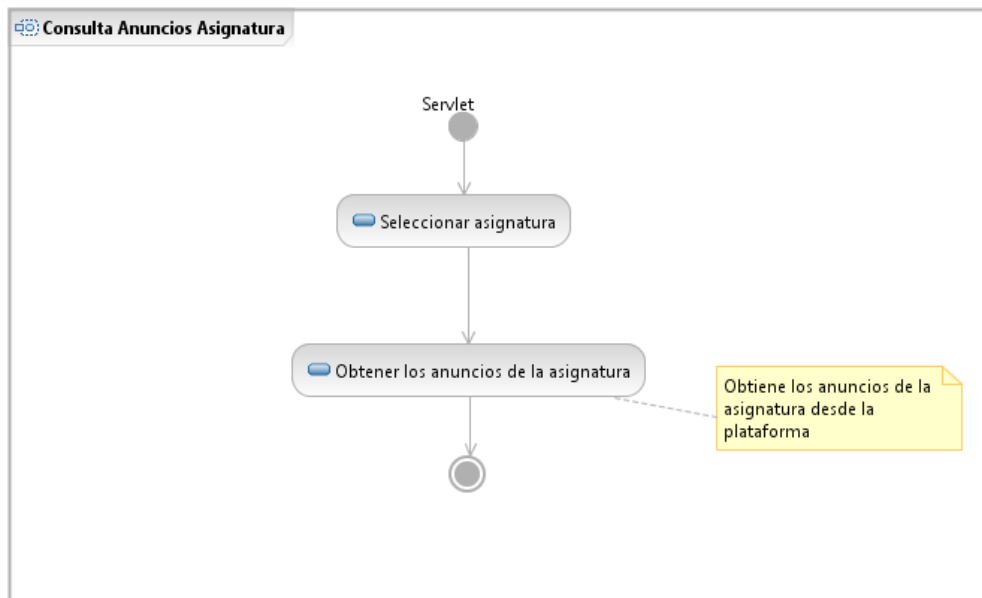


FIGURA 5.26 CASOS DE USO - ANUNCIOS

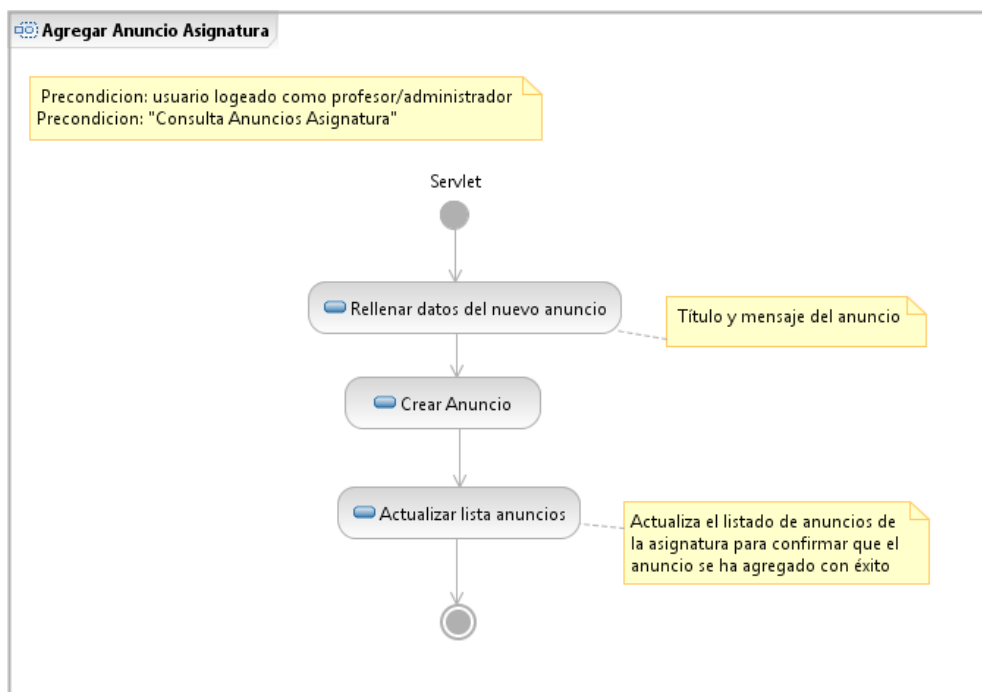


Caso de uso “Consulta Anuncios Asignatura”: obtiene el listado de anuncios de una asignatura.



**FIGURA 5.27 DIAGRAMA DE ACTIVIDAD – CONSULTA ANUNCIOS ASIGNATURA**

Caso de uso “Agregar Anuncio Asignatura”: añade un nuevo anuncio en una asignatura. Para ello, el usuario debe rellenar el título y el mensaje de dicho anuncio.



**FIGURA 5.28 DIAGRAMA DE ACTIVIDAD – AGREGAR ANUNCIO ASIGNATURA**

Caso de uso “Eliminar Anuncio Asignatura”: elimina el anuncio seleccionado de la asignatura. Para ello, se supone que ya ha seleccionado una asignatura y ha obtenido ya el listado de anuncios de la misma.

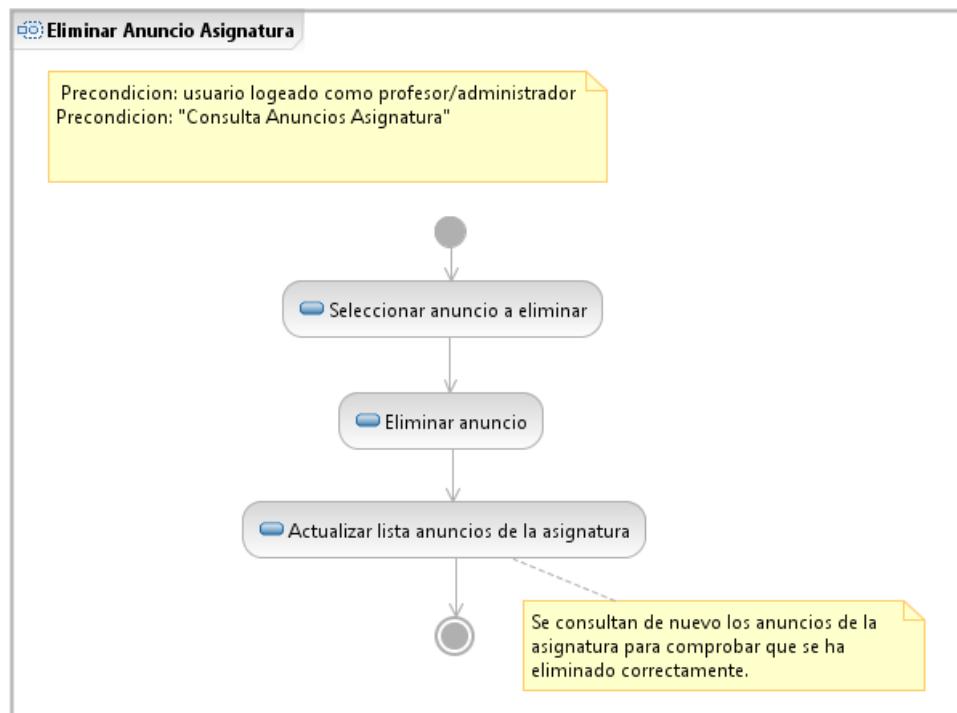


FIGURA 5.29 DIAGRAMA DE ACTIVIDAD – ELIMINAR ANUNCIO ASIGNATURA

## MÓDULO PLATAFORMA

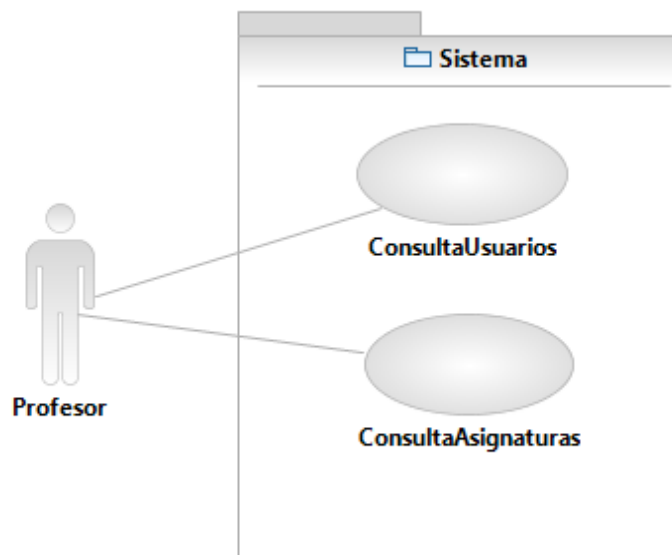
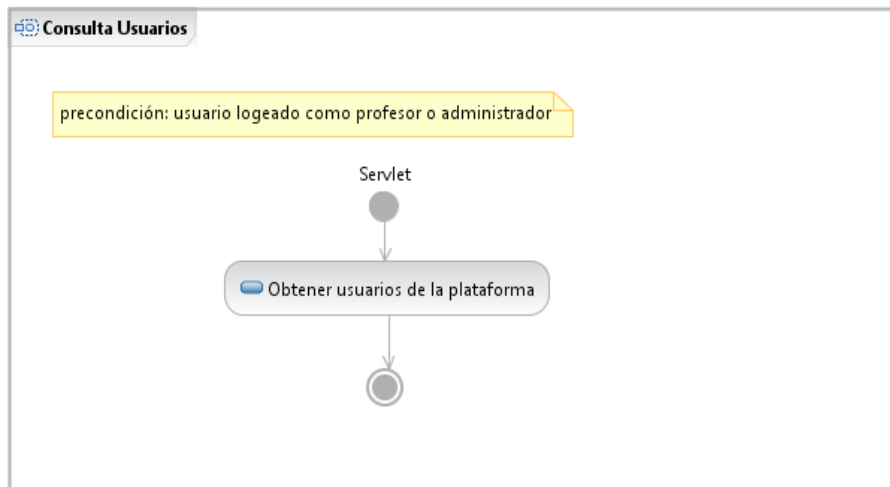


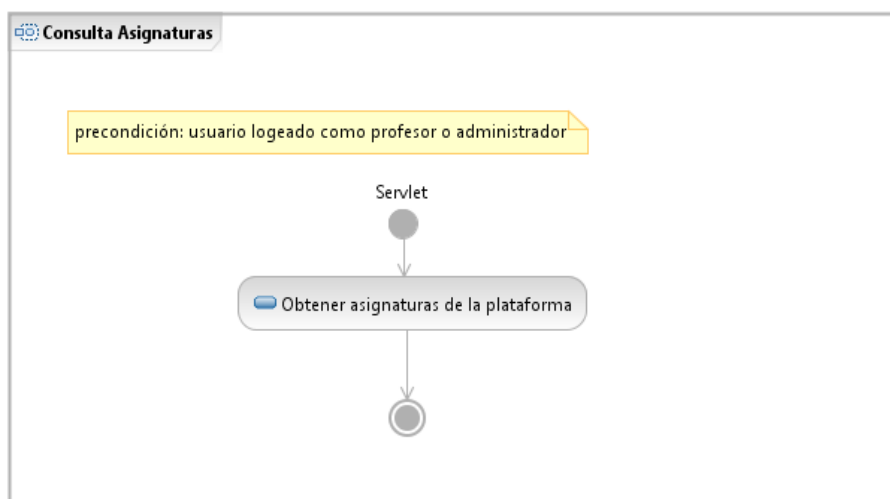
FIGURA 5.30 CASOS DE USO - PLATAFORMA

Caso de uso “Consulta Usuarios”: obtiene una lista con todos los usuarios dados de alta en la plataforma correspondiente del Campus Virtual.



**FIGURA 5.31 DIAGRAMA DE ACTIVIDAD – CONSULTA USUARIOS**

Caso de uso “Consulta Asignaturas”: obtiene una lista de todas las asignaturas virtualizadas en la plataforma correspondiente del Campus Virtual.



**FIGURA 5.32 DIAGRAMA DE ACTIVIDAD – CONSULTA ASIGNATURAS**

## CAPA DE PRESENTACIÓN

La capa de presentación es la que presenta el sistema del campus virtual al usuario, le comunica la información y captura la información del usuario.

Al haber optado por la utilización del framework de Struts para la implementación del patrón MVC (*Model View Controller*), es obligado el uso de JSPs (*Java Servlet Pages*) para representar la vista.

Las diferentes vistas con JSPs que se han desarrollado se muestran en la Figura 5.33.



FIGURA 5.33 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA PRESENTACIÓN

## CAPA DE LÓGICA DE NEGOCIO

La Capa de Lógica de Negocio del Campus Virtual es la encargada de recibir las peticiones del usuario y enviar las respuestas tras el proceso. En este caso, dichas respuestas son devueltas por los servicios web mediante la Capa de integración.

Además, es la encargada de comunicarse con la base de datos del campus virtual para su correcta gestión.

Para simplificar el diseño, esta capa se ha dividido en distintos paquetes:

1. Paquete **Actions**: este paquete contiene las acciones (*Actions*) encargadas de manejar las solicitudes del servlet Controlador de Struts y responder al cliente.



FIGURA 5.34 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (ACTIONS 1)

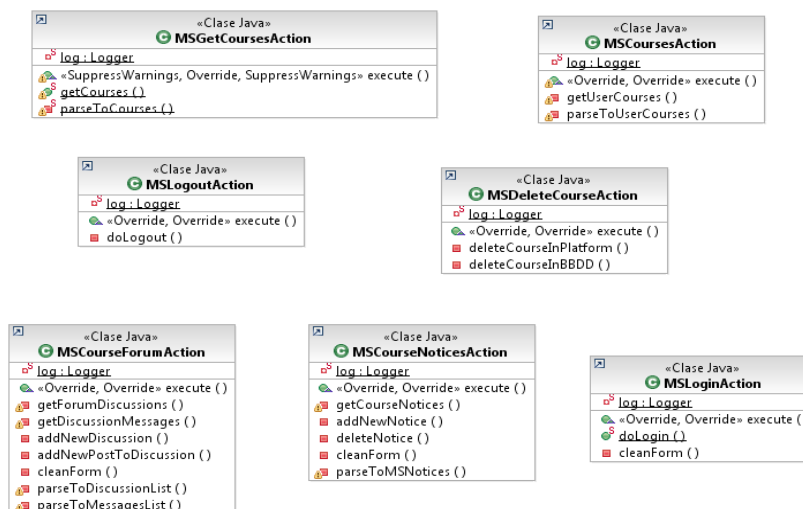


FIGURA 5.35 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (ACTIONS 2)

2. Paquete **Forms**: cada acción (*Action*) indicada arriba, lleva asociada un formulario que contiene los datos que dicha acción maneja. Para mayor legibilidad, mostramos sólo algunos de ellos, pudiendo encontrar el diagrama completo de clases en el proyecto UML diseñado para el Campus Virtual.

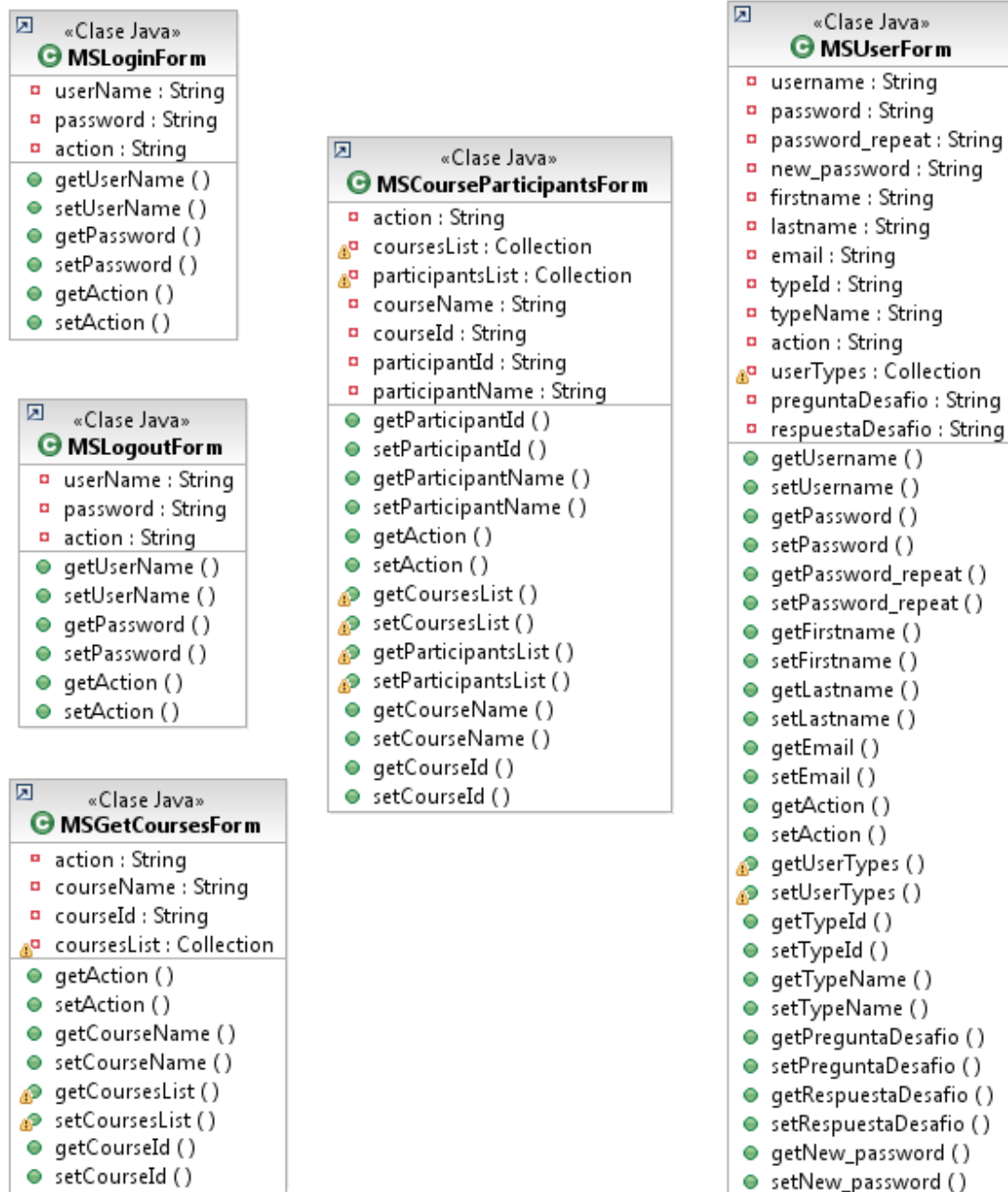


FIGURA 5.36 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (FORMS)

3. Paquete **Beans**: contiene los objetos o *beans* que utiliza el sistema del campus virtual. Para mayor legibilidad, mostramos algunos de ellos. El diagrama completo de clases se encuentra en el proyecto de UML diseñado para el Campus Virtual.

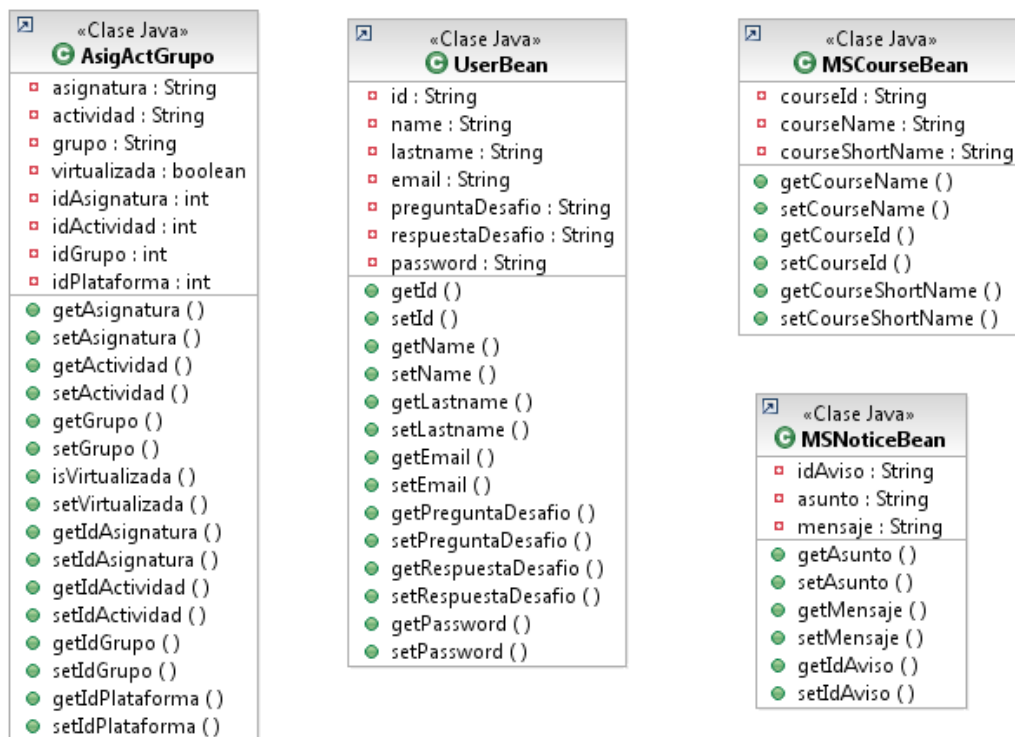


FIGURA 5.37 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (BEANS)

4. Paquete **BBDD**: aquí se encuentra la clase que accede a la base de datos desarrollada para el campus virtual.

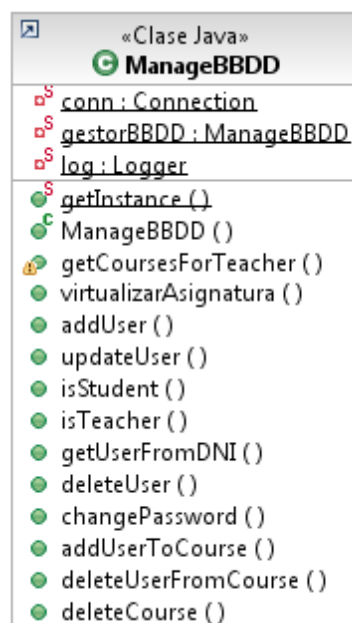


FIGURA 5.38 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (BBDD)

5. Paquete **Util**: en este último paquete se incluyen clases auxiliares utilizadas por la aplicación del Campus Virtual.

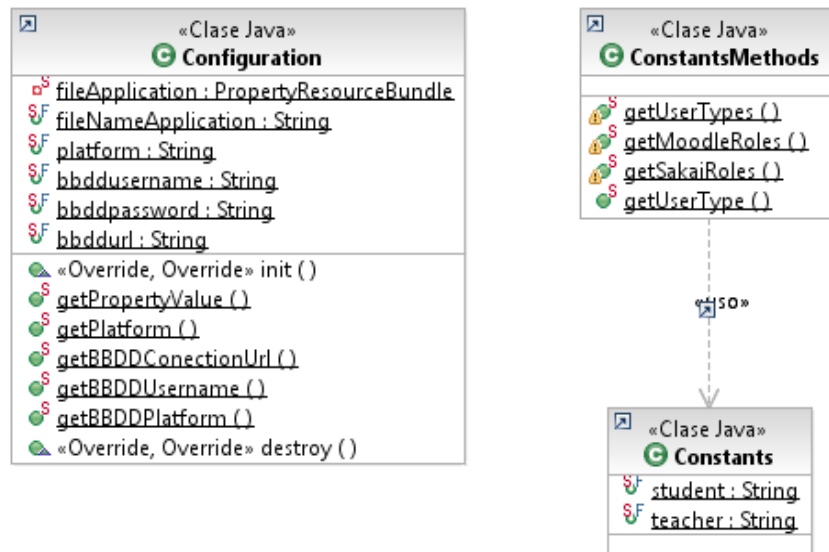


FIGURA 5.39 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA LOGICA NEGOCIO (UTIL)



## CAPA DE INTEGRACIÓN

En la capa de Integración se encuentran los objetos que interactúan con los distintos servicios web desarrollados para la plataforma del campus virtual. Es en esta capa de la arquitectura donde se ha implementado el patrón Factoría, como se ha mencionado anteriormente.

En la siguiente figura mostramos una parte de las clases para mayor legibilidad:



FIGURA 5.40 DIAGRAMA DE CLASES. CAMPUS VIRTUAL – CAPA INTEGRACION

El diagrama de clases completo se encuentra en el proyecto de UML diseñado para el Campus Virtual.

## DIAGRAMA DE SECUENCIA

A continuación, se muestra como ejemplo el diagrama de secuencia del Campus Virtual obtenido para el alta de una asignatura en el campus virtual.

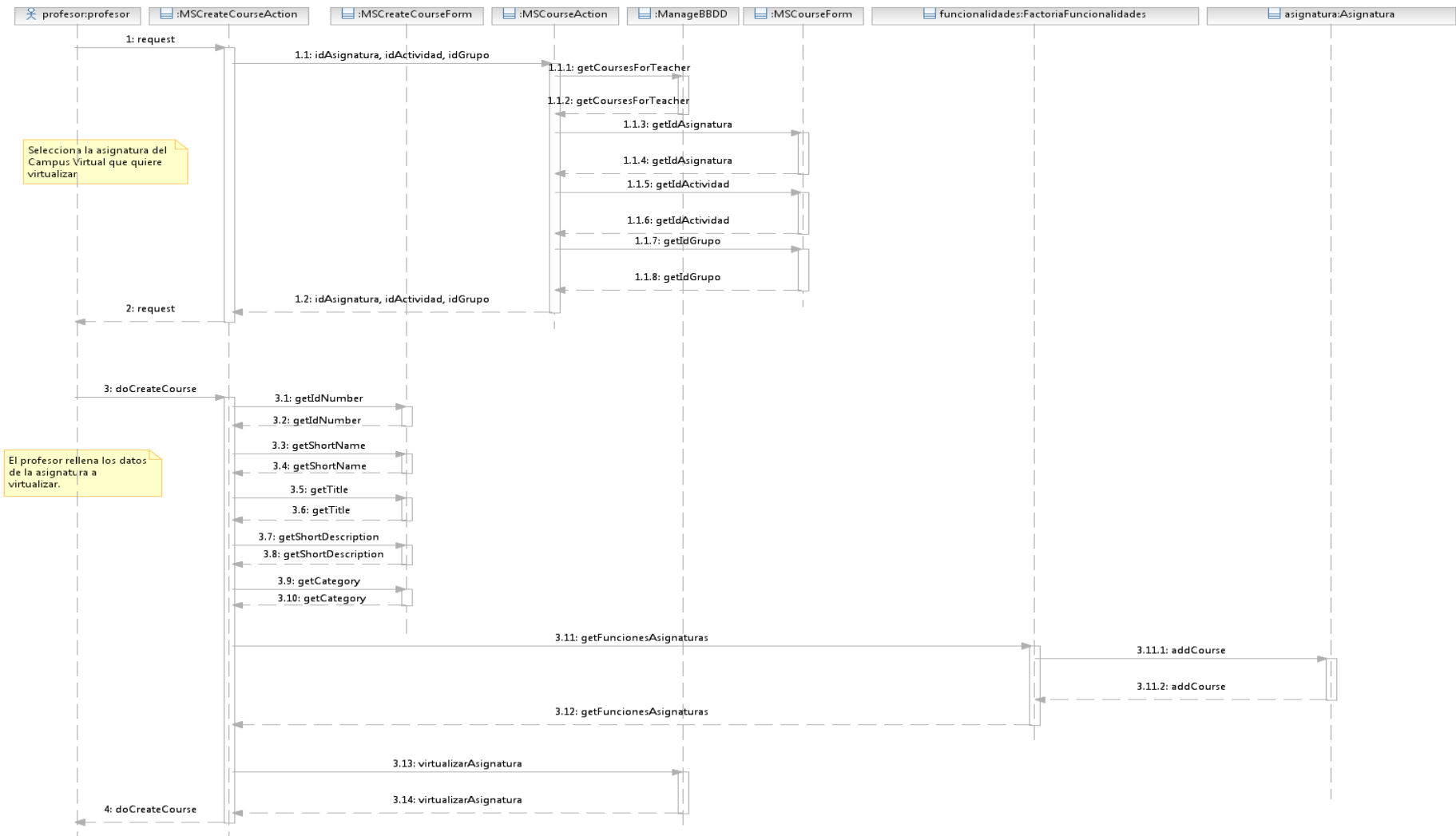


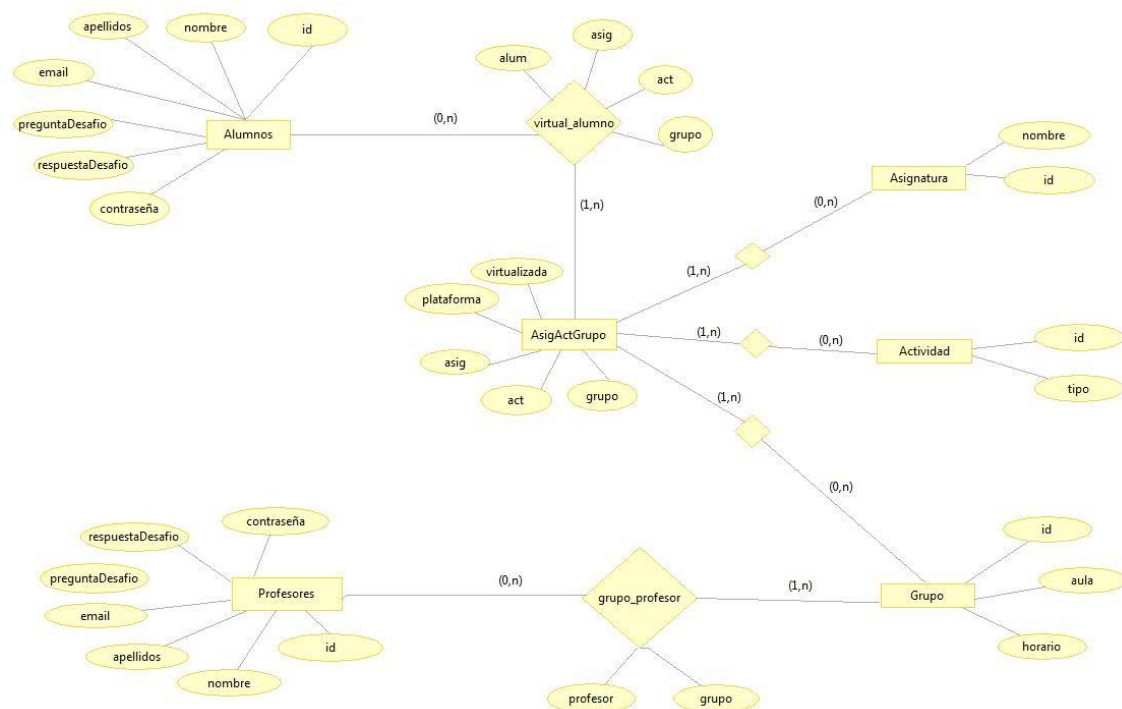
FIGURA 5.41 DIAGRAMA DE SECUENCIA. CAMPUS VIRTUAL – ALTA ASIGNATURA

El resto de diagramas de secuencia de la aplicación web del Campus Virtual se encuentran recogidas en el proyecto de UML diseñado para el Campus Virtual.

Nótese que la sencillez de la arquitectura del cliente nos ha llevado a obviar la presencia de servicios de la aplicación (Alur 03), utilizando directamente los action de Struts para implementar las funcionalidades. Arquitecturas más avanzadas sí que utilizarían dichos servicios de aplicación.

### 5.3.3 DIAGRAMA ENTIDAD RELACIÓN

A continuación se muestra el diagrama de entidad – relación que representa conceptualmente el esquema de la base de datos diseñada para el Campus Virtual.



**FIGURA 5.42 DIAGRAMA DE ENTIDAD RELACION**

Describimos brevemente la utilidad de cada tabla aquí propuesta:

- **ALUMNOS:** esta tabla almacena la información de los alumnos dados de alta en el Campus Virtual. El campo “id” es el identificador único de cada alumno, y se corresponde con su DNI.
- **PROFESORES:** almacena la información de los profesores agregados en el Campus Virtual. Es similar a la tabla ALUMNOS, pero se ha decidido separarla para mantener un control sobre la virtualización de asignaturas, que solo puede ser realizada por los profesores.
- **ASIGNATURA:** contiene la lista de asignaturas del Campus Virtual
- **ACTIVIDAD:** contiene la lista de actividades que puede tener cada asignatura del Campus Virtual
- **GRUPO:** los grupos disponibles para las asignaturas del Campus Virtual

- GRUPO\_PROFESOR: aquí se guarda la relación de los grupos a los que un profesor pertenece.
- VIRTUAL\_ALUMNO: guarda la lista de las asignaturas virtualizadas de un usuario.
- ASIGACTGRUPO: cada asignatura a virtualizar en la plataforma está definida por la asignatura, la actividad y el grupo a la que pertenece. Así, podemos tener, por ejemplo, la asignatura LP3 virtualizada para la Actividad de teoría en el Grupo A. Además, indica si está virtualizada, y en qué plataforma (esto último de manera temporal por necesidades del proyecto)

## 5.4 DISEÑO SERVICIOS WEB

### 5.4.1 MOODLE WS

Antes de pasar a describir el diseño del servicio web hemos de decir que al estar implementado en PHP se pierde la orientación a objetos. Para poder realizar el diseño mediante una herramienta CASE hemos tenido que admitir los archivos de PHP con el código del servicio web y las librerías de la plataforma Moodle como clases, y las funciones que allí se implementan como sus métodos.

En nuestro proyecto utilizamos la versión 2.0 de la plataforma Moodle en la que existe un modulo que proporciona una interfaz de interacción con la plataforma mediante servicios web. El problema es que al estar en fase de desarrollo este modulo está incompleto y no es recomendada su utilización en producción. En el momento en que comenzamos este proyecto las funciones que ofrecía el modulo eran escasas y su estructura no estaba clara. Al ser una plataforma de software libre la documentación y el soporte son limitados. Por estas cuestiones decidimos desarrollar desde cero los servicios web para Moodle. A día de hoy el desarrollo del módulo de servicios web está bastante más desarrollado que al principio del proyecto por lo que sería interesante evaluar una posible fusión de los servicios ya realizados con los que proporcione la plataforma.

De esta manera hemos creado seis servicios diferentes que ofrecen un total de 25 operaciones que interactúan con la plataforma. A continuación se describen los diagramas de clases de los servicios nombrados organizados por capas.

Los diferentes servicios están implementados mediante seis archivos PHP. Como ya se ha mencionado, a la hora de diseñar los diagramas de clases hemos considerado a cada uno de estos archivos como una clase y a las operaciones que ofrecen como sus métodos. Para implementar estas operaciones los servicios hacen uso de una capa de integración que esta compuestas por librerías de la propia plataforma. Estas librerías son las que contienen las funciones que están especificadas en la API.

Durante la implementación de los servicios web encontramos ciertos problemas a la hora de utilizar ciertas funciones de la API de Moodle sobre todo en la gestión de archivos. Por ello tuvimos que reimplementar dos funciones eliminando las líneas que nos daban problemas, todo ello sin perder funcionalidad. Estas funciones son `get_file_info_course_content1()` y `get_file_storage2()`.

## CAPA DE LÓGICA DE NEGOCIO

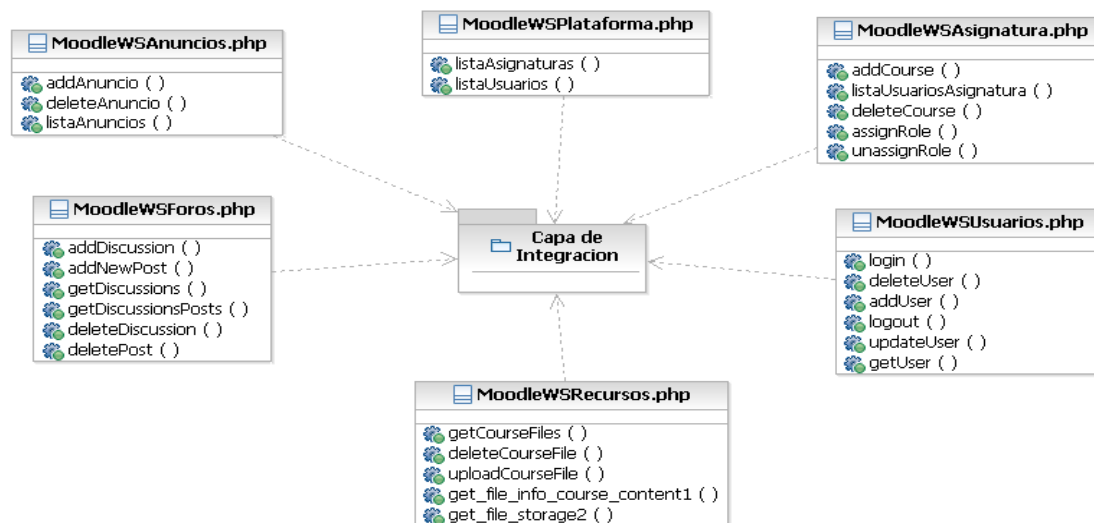


FIGURA 5.43 DIAGRAMA DE CLASES. MOODLE WS – CAPA LOGICA NEGOCIO

## CAPA DE INTEGRACIÓN

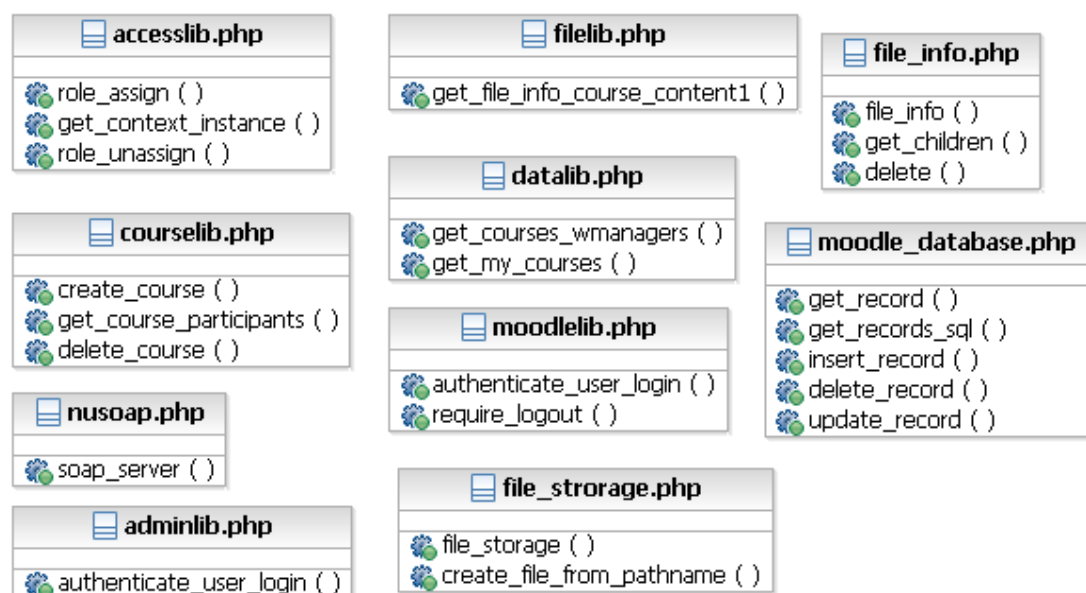


FIGURA 5.44 DIAGRAMA DE CLASES. MOODLE WS – CAPA INTEGRACION

## DIAGRAMA DE SECUENCIA

A continuación se muestra como ejemplo el diagrama de secuencia de los servicios web de Moodle para el alta de una asignatura en la plataforma.



**FIGURA 5.45 DIAGRAMA DE SECUENCIA. MOODLE WS – ALTA ASIGNATURA**

El resto de diagramas de secuencia se encuentran en el proyecto de UML diseñado para los servicios web de Moodle.

#### 5.4.2 SAKAI WS

Para el correcto funcionamiento del servicio web de Sakai (SakaiScript), resulta indispensable la existencia de una capa superior, que se representa como los servicios web SakaiWS, se requieren para la modulación de las funcionalidades (Usuarios, Asignaturas, Recursos, Foros, Anuncios y Plataforma), ya que esta labor no se pudo realizar a partir del servicio web modificable que proporciona Sakai 2.6; a su vez estos servicios web actúan de intermediarios entre el campus virtual y la plataforma subyacente Sakai 2.6, permitiendo mantener la igualdad de parámetros entre las distintas plataformas, ya que son los servicios web SakaiWS los encargados de distinguir los parámetros que requiere el servicio web de la plataforma.

Se han utilizado dos servicios web de la plataforma, SakaiLogin y SakaiScript, diferenciando que SakaiScript permite su modificación, por este motivo, es en SakaiScript donde se han desarrollado las funcionalidades necesarias para el funcionamiento actual del campus virtual de marca blanca.

Aproximadamente, para este proyecto se implementaron un 65% de las funcionalidades y la plataforma ofrece el 35% restante, de este último porcentaje se distingue un 15% de funcionalidades reimplementadas debido a la dificultad existente para la modificación de los permisos de los distintos tipos de usuario y la necesidad de la pertenencia del administrador de la plataforma a todos los cursos que se creen; el 20% restante corresponde funcionalidades que dispone la plataforma y que han sido utilizadas sin modificaciones.

La distribución de las funcionalidades disponibles, y sus correspondientes funciones y referencias, se especifican en las tablas siguientes:

Funcionalidades utilizadas

Función del servicio web SakaiWS	Función del servicio web de Sakai	Utilidad/Funcionalidad	Modulo	
login	login	Acceso	Usuarios	Esta operación está alojada en el servicio web SakaiLogin.jws, no modificable.
logout	logout	Salida	Usuarios	Esta operación está alojada en el servicio web SakaiLogin.jws, no modificable.
addUser	addNewUser	Alta Alumno / Alta Profesor	Usuarios	
deleteUser	removeUser	Baja Alumno / Baja Profesor	Usuarios	
deleteCourse	removeSite	Baja Asignatura	Asignaturas	Se utiliza borrarAsignatura, en lugar de esta operación por seguir la coherencia entre nombres, pero esta operación es un copia exacta de removeSite
assignRole	addMemberToSiteWithRole	Alta Alumno Asignatura / Alta Profesor Asignatura	Asignaturas	
unassignRole	removeMemberToSite	Baja Alumno Asignatura / Baja Profesor Asignatura	Asignaturas	

# Funcionalidades desarrolladas

Función del servicio web SakaiWS	Función del servicio web de Sakai	Utilidad/Funcionalidad	Módulo
updateUser	modificarDatos	Modificar Datos / Modificar Contraseña	Usuarios
getUser	mostrarDatos	ModificarDatos	Usuarios
listaAsignaturasUsuarios	listaAsignaturasUsuarios	Consulta Asignaturas Usuario	Usuarios
addCourse	crearForo	Alta Asignatura	Asignaturas
listaUsuariosAsignatura	listaUsuariosAsignatura	Consulta Usuarios Asignatura	Asignaturas
listaAnuncios	listaAnuncios	Consulta Anuncios Asignatura	Anuncios
addAnuncio	agregarAnuncio	Agregar Anuncio Asignatura	Anuncios
deleteAnuncio	borrarAnuncio	Eliminar Anuncio Asignatura	Anuncios
addDiscussion	agregarTemaForo	Agregar Tema Foro	Foro
deleteDiscussion	borrarTemaForo	Eliminar Tema Foro	Foro
addNewPost	agregarMensajeForo	Agregar Mensaje Tema	Foro
deletePost	borrarMensajeForo	Eliminar Mensaje Tema	Foro
getDiscussions	listadoForosyTemas	Consulta Temas Foro Asignatura	Foro
getDiscussionPosts	listaMensajesTemaForo	Consulta Mensajes Tema	Foro
getCourseFiles	getRecursosAsignatura	Consulta Recursos Asignatura	Recursos
Upload	addRecursosAsignatura	Agregar Recurso Asignatura	Recursos
deleteCourseFile	removeRecursosAsignatura	Eliminar Recurso Asignatura	Recursos
listaAsignaturas	getListaAsignaturas	Consulta Asignaturas	Plataforma
listaUsuarios	getListaUsuarios	Consulta Usuarios	Plataforma



Funcionalidades desarrolladas en base a otras

Función del servicio web SakaiWS	Función del servicio web de Sakai	Función base	Utilidad/Funcionalidad	Módulo	
addCourse	agregarAsignatura	addNewSite	Alta Asignatura	Asignaturas	Los permisos de cada tipo de usuario difieren entre el campus virtual de marca blanca y la plataforma subyacente.
addCourse	nuevaPaginaAsignatura	addNewPageToSite	Alta Asignatura	Asignaturas	Los permisos de cada tipo de usuario difieren entre el campus virtual de marca blanca y la plataforma subyacente.
addCourse	nuevaHerramientaPagina	addNewToolToPage	Alta Asignatura	Asignaturas	Los permisos de cada tipo de usuario difieren entre el campus virtual de marca blanca y la plataforma subyacente.
addCourse	addMemberToSite	addMemberToSiteWithRole	Alta Asignatura	Asignaturas	Debido a la necesidad de que el administrador pertenezca a todos los cursos (asignaturas), se crea esta función en base a addMemberToSiteWithRole.

**TABLA 3. FUNCIONALIDADES DESARROLLADAS EN SAKAIWS**

## CAPA DE LÓGICA DE NEGOCIO



FIGURA 5.46 DIAGRAMA DE CLASES. SAKAI WS – CAPA LOGICA NEGOCIO

## CAPA DE INTEGRACIÓN

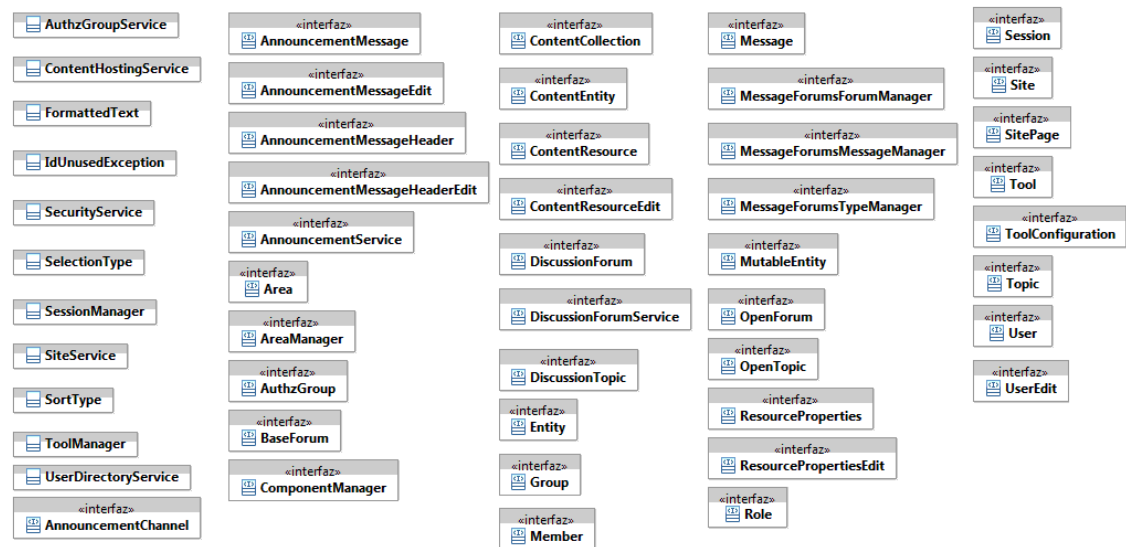
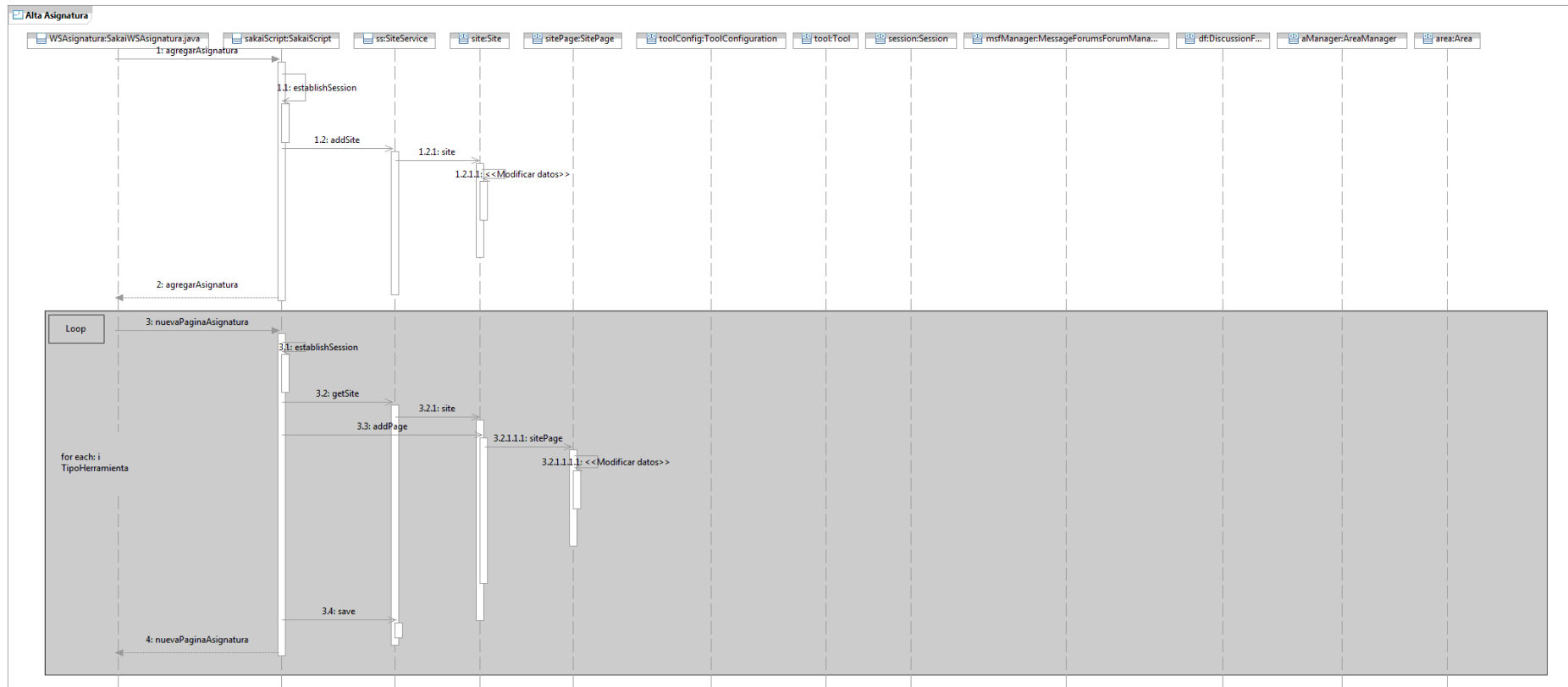
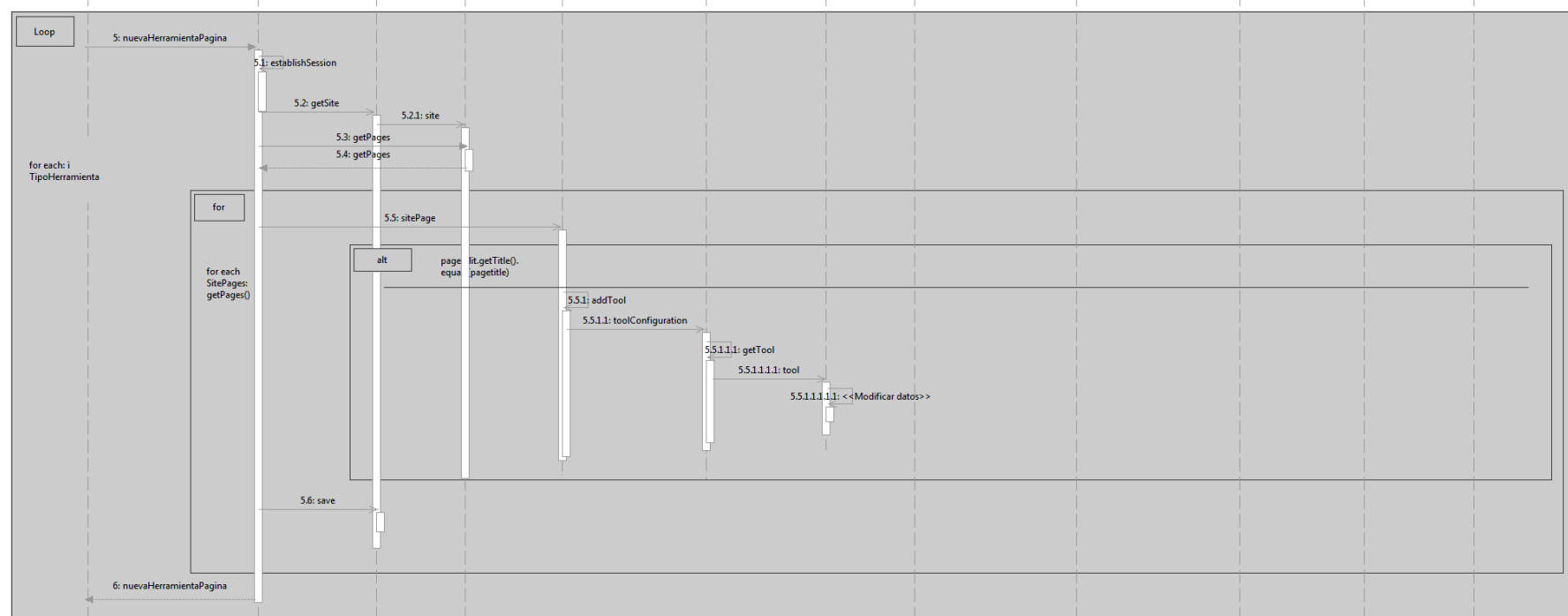


FIGURA 5.47 DIAGRAMA DE CLASES. SAKAI WS – CAPA INTEGRACION

## DIAGRAMA DE SECUENCIA

En la Figura 5.48 se muestra como ejemplo el diagrama de secuencia de los servicios web de Sakai para el alta de una asignatura. El resto de diagramas de secuencia se encuentran en el proyecto de UML diseñado para los servicios web de Sakai.





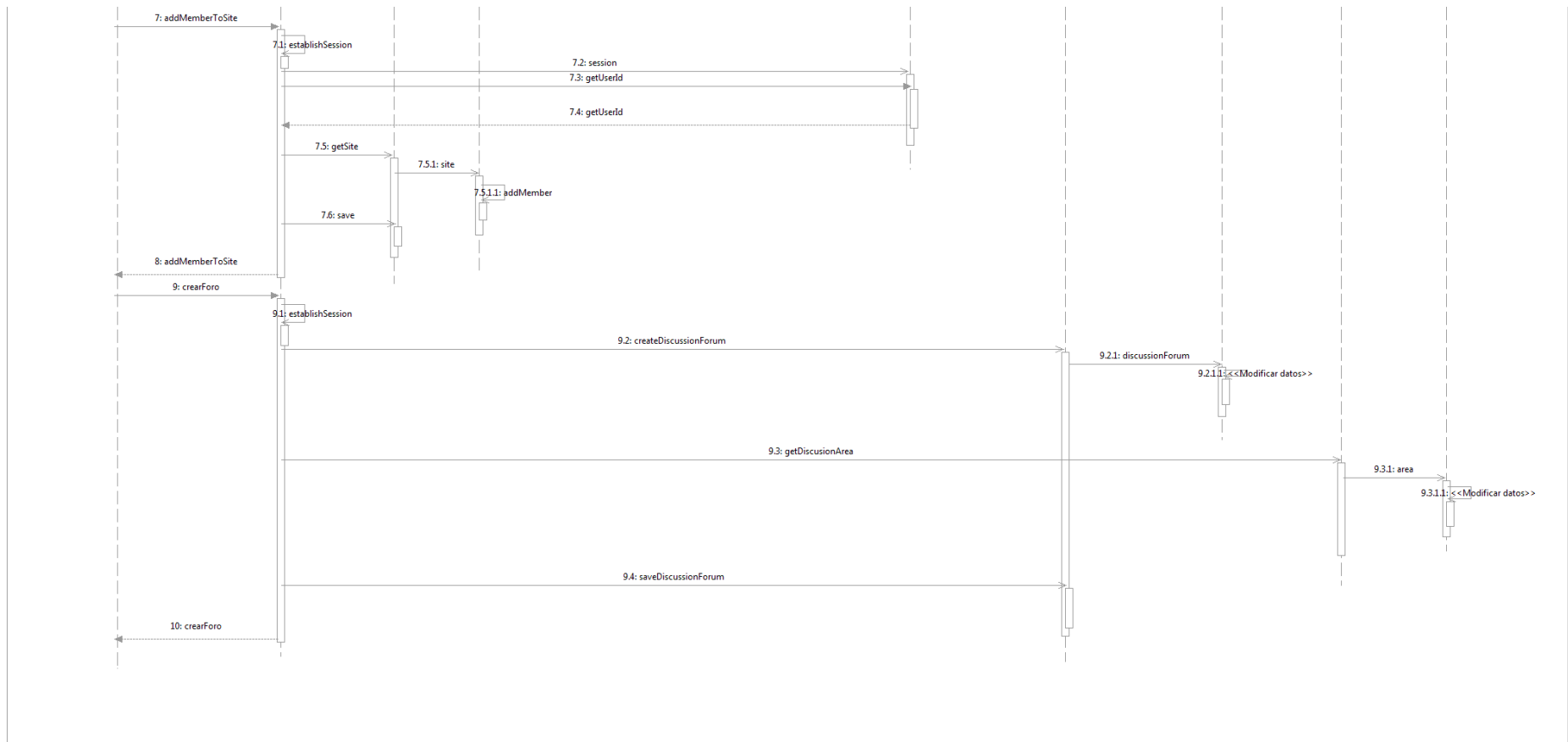


FIGURA 5.48 DIAGRAMA DE SECUENCIA. SAKAI WS – ALTA ASIGNATURA

## 6. CONCLUSIONES Y TRABAJO FUTURO

### APORTACIONES A NUESTRA FORMACIÓN

La principal aportación de este trabajo ha sido la de utilizar gran parte de los conocimientos adquiridos a lo largo de nuestros estudios en el contexto de un proyecto de cierta envergadura desarrollado en equipo. Además, la interacción con el director del proyecto, hasta cierto punto, ha servido para simular una interacción con un cliente real. Dicha interacción ha sido aún más compleja si cabe, ya que el director del proyecto no sólo proponía requisitos del sistema, sino además, requisitos técnicos sobre la implementación de dicho sistema.

El método que hemos seguido para desarrollar nuestro proyecto ha sido el de ir cumpliendo una serie de entregas parciales e hitos en nuestra planificación. En este sentido se ha seguido un modelo de proceso incremental. Para realizar este seguimiento hemos mantenido reuniones periódicas con el profesor director en su despacho en las que definíamos los pasos a seguir hasta la siguiente reunión. Hemos dedicado también muchas horas de trabajo en común, pero la mayoría del trabajo lo hemos hecho individualmente lo cual debido a la naturaleza del proyecto no nos ha causado demasiados problemas.

Durante el desarrollo de nuestro proyecto hemos manejado diferentes herramientas y tecnologías. Estábamos familiarizados con el funcionamiento de algunas de ellas, otras sin embargo nos eran totalmente desconocidas por lo que la labor de aprendizaje durante los primeros meses fue de gran intensidad.

Esta primera fase del proyecto consistió en conocer las bases de la arquitectura SOA y su implementación mediante Servicios Web. En concreto tuvimos que aprender cómo funciona el protocolo SOAP, especialmente el formato de los mensajes XML y la estructura de fichero de descripción del servicio, WSDL. Así mismo, fue necesario estudiar cómo se desarrollaban los Servicios Web tanto en Java como en PHP, pues hay diferencias entre ambas. Por tanto, el proyecto ha demostrado nuestra capacidad de aprendizaje autónomo.

Por otro lado tuvimos que analizar a fondo las dos plataformas para las que desarrollamos los servicios. No existe documentación específica sobre las API de las dos plataformas por lo que nos ha llevado mucho tiempo conseguir desarrollar las operaciones que actúen sobre las plataformas. De hecho, mantuvimos una reunión con técnicos responsables del Campus Virtual de la UCM, que nos ayudaron a entender cómo estaban organizadas y desarrolladas las plataformas de Sakai y Moodle, y cómo interactuaban con las limitadas APIs de cada una de ellas.

En el caso del Servicio Web que interactúe con Sakai ha de estar implementado en Java. Para Moodle la implementación ha sido en PHP para poder acceder a las funciones de la API. Ambos Servicios deben estar implantados en la misma máquina en la que están desplegadas las plataformas.

Para implementar el Campus Virtual que consume estos Servicios Web hemos utilizado el Framework Struts creado por Apache para J2EE. Con la utilización de Struts hemos reducido considerablemente el tiempo de desarrollo de la aplicación. También hemos utilizado varios patrones de diseño, como Singleton o Factoría Abstracta.

Los mayores problemas los hemos encontrado a la hora de crear los Servicios Web. Dado la complejidad y la falta de documentación de las plataformas, el acceso y dominio de las funciones proporcionadas por la API ha sido de gran dificultad.

## CONCLUSIONES DEL TRABAJO

Las funciones que se han desarrollado para la integración de ambas plataformas y que soporta actualmente el Campus Virtual son las siguientes:

- Login: validación del usuario en la plataforma, que permite acceder al resto de operaciones del Campus Virtual.
- Gestión de usuarios: permite dar de alta usuarios en la plataforma y la BBDD del Campus Virtual, modificar los datos personales de los mismos, y borrarlos.
- Gestión de asignaturas: permite virtualizar asignaturas en la plataforma, y borrarlas.
- Alta y baja de participantes en asignaturas
- Gestión de archivos de cada asignatura, permitiendo agregar archivos, su consulta, y su eliminación.
- Gestión de anuncios, para crear y eliminar anuncios en una asignatura
- Gestión de foros, permitiendo a los usuarios participar en los foros de una asignatura, con la posibilidad de crear temas y mensajes nuevos.
- Logout: desconexión de la plataforma

Así pues, se puede observar que el objetivo del proyecto se ve satisfecho con la solución desarrollada por el grupo, al contar con un Campus Virtual de marca blanca que puede integrarse con las plataformas de Moodle 2.0 y Sakai 2.6 mediante Servicios Web. Además, el Campus Virtual está integrado con la BBDD y la plataforma al mismo tiempo, manteniendo los datos entre ambos sincronizados. La plataforma final seleccionada es transparente al usuario.

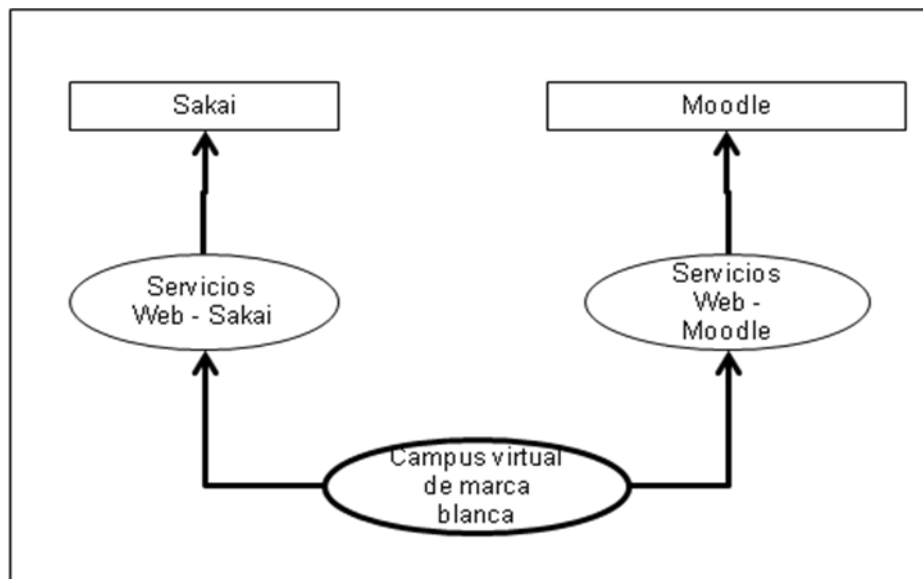


FIGURA 6.1 CONCLUSION



## TRABAJO FUTURO

En primera instancia se pueden añadir más funcionalidades al Campus Virtual todavía no implementadas, como por ejemplo la gestión de calificaciones, exámenes o encuestas, la fusión de asignaturas, etc. Es importante tener en cuenta que las funcionalidades a desarrollar deben estar disponibles en todas las plataformas que deseemos integrar.

Otra vía de desarrollo consistiría en la integración de este Campus Virtual de marca blanca con otras plataformas LMS. Para ello, sería necesario crear otros Servicios Web que accedieran a dichas plataformas y soportaran todas las operaciones ya existentes por el Campus Virtual. Además, estos servicios han de integrarse en la aplicación web del Campus Virtual para que pueda hacer uso de ellos.

También es importante destacar el desarrollo necesario de una interfaz de usuario más amigable y simplificada, puesto que en este proyecto se provee de una interfaz sencilla. No se ha desarrollado una interfaz de usuario más avanzada por las limitaciones temporales y porque el núcleo del proyecto era el desarrollo de los servicios web implementados.

El desarrollo de mecanismos de traducción entre los contenidos desplegados en las dos plataformas de e-learning sería otro trabajo muy interesante a llevar a cabo.

Otro aspecto a tener en cuenta sería la exportación de los servicios del campus virtual como servicios web. De esta forma el campus virtual podría integrarse fácilmente con un sistema de gestión universitario externo.

En la medida que todo este trabajo futuro se desarrollase, podríamos hablar de la consecución de un campus virtual de nueva generación. Precisamente, este proyecto de Sistemas Informáticos sienta las bases para el desarrollo de dicho campus virtual.



## 7. BIBLIOGRAFÍA

- (ADL 10) Advanced Distributed Learning. <http://www.adlnet.gov/Pages/Default.aspx>
- (Allison 08) D.H. Allison, P.B. DeBlois. and the EDUCAUSE CIC. Current IT Issues Survey Report, 2008. *EDUCAUSE Quarterly*, vol. 3, n.2
- (Alur 03) Alur D., Crupi J., Malks D. Core J2EE Design Patterns. 2nd edition. Prentice Hall/Sun Microsystems Press, 2003.
- (AOS PHP) Patrones de Diseño de Servicios Web en PHP. Javier Castagnetto. <http://www.slideshare.net/jesus.castagnetto/construyendo-una-aos-con-php-patrones-de-diseo-de-servicios-web-en-php-presentation>
- (APB 09) Blackboard 9: ahora Blackboard Learn <http://aprendizajedistancia.blogspot.com/2009/02/blackboard-9-ahora-blackboard-learn.html>
- (BlackBoard 10) BlackBoard. <http://www.blackboard.com/Teaching-Learning/Learn-Platform.aspx>
- (BlackBoard DS 9.1) Our Next Generation Teaching and Learning Platform [http://www.blackboard.com/resources/learn/Bb\\_Learn\\_91\\_Datasheet\\_Classic.pdf](http://www.blackboard.com/resources/learn/Bb_Learn_91_Datasheet_Classic.pdf)
- (Carliner 05) Carliner, Course Management Systems vs. Learning Management Systems [http://www.astd.org/LC/2005/1105\\_carliner.htm](http://www.astd.org/LC/2005/1105_carliner.htm), 2005.
- (Confluence 10) <http://issues.sakaiproject.org/confluence/>
- (Dewey 06) Dewey, B.I., DeBlois, P.B. and the EDUCAUSE Current Issues Committee (2006) Current IT Issues Survey Report, 2006. *EDUCAUSE Quarterly*, 29 (2) <http://www.educase.edu/apps/eq/eqm06/eqm0622.asp>, 2006.
- (E-learning 02) E-learnign. <http://www.maestrosdelweb.com/editorial/elearning/>
- (EduTools 10) <http://www.edutools.info>
- (Epper 04) Epper, R. M., Garn, M. The Virtual University in America: Lessons from Research and Experience. EDUCAUSE Centre for Applied Research (ECAR) Research Bulletin, 2004(2) <http://www.educause.edu/LibraryDetailPage/666?ID=ERB0402>, 2004.
- (Erl 05) Erl, T. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall 2005.
- (Fowler 03) Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2003.
- (Gamma 06) Gamma, E., Helm, R., Johnson, R., Vlissides, J. Patrones de Diseño: Elementos de Software Orientado a Objetos Reutilizables. Addison-Wesley, 2006.
- (Green 05) Green, K.C. The 2005 National Survey of Information Technology in U.S. Higher Education: Growing Campus Concern about IT Security; Slow Progress on IT Disaster Planning, The Campus Computing Project, <http://www.campuscomputing.net>, 2005.
- (Grupo CMC 10) SOA. ¿Qué es? ¿Para quién sirve? ¿Quién lo utiliza? José Vicente Hernández. [http://www.grupocmc.es/articulos/soa\\_jvHernandez.pdf](http://www.grupocmc.es/articulos/soa_jvHernandez.pdf)

(Guri-Rosenblit 05) Guri-Rosenblit, S. Eight Paradoxes in the Implementation Process of E-learning in Higher Education. Higher Education Policy, 18, 5-29, 2005.

(Historia WS 05) Historia de los Web Services,  
<http://www.desarrolloweb.com/articulos/1883.php>

(Hohpe 03) Hohpe G., Wolf, B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, 2003.

(Hohpe 07) Hohpe, G. SOA Patterns -- "New Insights or Recycled Knowledge?". Fifth International Workshop on SOA and Web Services. Canadá, 2007.

(IMS 05) Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning  
<http://www.biblioweb.dgsca.unam.mx/libros/repositorios/estandares.htm#54r>

(IMS 09) IMS Abstract Framework: Glossary,  
<http://www.imsglobal.org/af/afv1p0/imsafglossaryv1p0.html>, 2009.

(IMS 10) IMS Global Learning Consortium <http://www.imsglobal.org/>

(JSF) Introducción a la Tecnología JavaServer Faces  
[http://www.programacion.com/articulo/introduccion\\_a\\_la\\_tecnologia\\_javascript\\_faces\\_233](http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233)

(Kaplan-Leison 09) E. Kaplan-Leison, ASTD Learning Circuits. Glossary  
<http://www.learningcircuits.org/glossary>, 2009.

(Leidner 95) Leidner, D. E., Jarvenpaa, S. L. The use of information technology to enhance management school education: a theoretical view. MIS Quarterly 19 (3), 265-291, 1995.

(Moodle 10) Moodle <http://moodle.org>

(Moodle vs Sakai 10) Moodle vs. Sakai, <http://moodle-vs-sakai.blogspot.com/>

(Navarro 07) A. Navarro and A. Fernández-Valmayor, "Conceptualization of Hybrid Websites", Internet Research vol. 17, 2007, pp. 207-228.

(Navarro 08a) A. Navarro, A. Fernández-Valmayor, B. Fernandez-Manjón, and J.L. Sierra "Characterizing navigation maps for web applications with the NMM approach", Sci. of Comp. Program., vol. 71, 2008, pp. 1-16.

(Navarro 08b) A. Navarro, J. Merino, A. Fernández-Valmayor, and J. Cristobal, "Translating Workflow Diagrams into Web Designs", Proc. SEKE 2008, 2008, pp. 667-672.

(Navarro 10) Towards a New Generation of Virtual Campuses  
Antonio Navarro, Jesús Cristóbal, Alfredo Fernández-Valmayor, Carmen Fernández, Héctor Hernanz, Salvador Guillomía, and Félix Buendía. AICT 2010, 2010.

(Neville 05) Neville, K., Heavin, C., Walsh, E. A Case in Customizing E-Learning. Journal of Information Technology 20 (2), 117-129, 2005.

(Newcomer 05) Newcomer E., Lomow G. Understanding SOA with Web Services. Addison Wesley, 2005.

(OMG 01) OMG MDA Guide Version 1.0.1, <http://www.omg.org/cgi-bin/doc?omg/03-06-01>, 2001

(PHC Consultores 10) Capacitación docente.

[http://pchconsultores.com/joomla/index.php?option=com\\_content&view=article&id=52&Itemid=30](http://pchconsultores.com/joomla/index.php?option=com_content&view=article&id=52&Itemid=30)

(PLS 04) PLS RAMBOLL Management for the European Commission. Studies in the Context of the E-learning Initiative: Virtual Models of European Universities. [http://www.elearningeuropa.info/extras/pdf/virtual\\_models.pdf](http://www.elearningeuropa.info/extras/pdf/virtual_models.pdf), 2004.

(Real World SOA). La arquitectura orientada a servicios (SOA) de Microsoft aplicada al mundo real. [http://www.google.es/url?sa=t&source=web&cd=3&ved=0CCoQFjAC&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2Fc%2F2%2Fc%2Fc2ce8a3a-b4df-4a12-ba18-7e050aef3364%2F070717-Real\\_World\\_SOA.pdf&ei=GikaTIDdAYP14AbRs-GyCg&usg=AFQjCNEiHNL4JcX\\_nkSCAoJGxlg45ENOUA&sig2=HOuNiyEzh31B5YuSojc1Xw](http://www.google.es/url?sa=t&source=web&cd=3&ved=0CCoQFjAC&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2Fc%2F2%2Fc%2Fc2ce8a3a-b4df-4a12-ba18-7e050aef3364%2F070717-Real_World_SOA.pdf&ei=GikaTIDdAYP14AbRs-GyCg&usg=AFQjCNEiHNL4JcX_nkSCAoJGxlg45ENOUA&sig2=HOuNiyEzh31B5YuSojc1Xw)

(RJE 10) Plataforma de cursos virtuales

[http://recursostic.javeriana.edu.co/atico/index.php?option=com\\_content&task=view&id=32&Itemid=74](http://recursostic.javeriana.edu.co/atico/index.php?option=com_content&task=view&id=32&Itemid=74)

(Sakai 10) Sakai Project <http://sakaiproject.org/>

(SCORM 10) SCORM for LMS <http://www.scorm.com/>

(SOA JavaEE 09) Implementing SOA Using Java™ EE. B. V. Kumar; Prakash Narayan; Tony Ng . Prentice Hall, 2009.

(UCM VC 09) UCM Virtual Campus, <http://www.ucm.es/campusvirtual>, 2009.

(UVG) Tutorial Sakai. Entorno Virtual de Enseñanza-Aprendizaje. Universidad del Valle de Guatemala. <http://www.slideshare.net/hguisela/tutorial-sakai>

(Van Dusen 97) Van Dusen, Gerald C. The Virtual Campus: Technology and Reform in Higher Education. ASHE-ERIC Higher Education Report Volume 25, No. 5. Washington, D.C.: The George Washington University, Graduate School of Education and Human Development, 1997.

(Whittington 98) Whittington, C.D., Sclater, N. Building and Testing a Virtual University. Computers & Education 30 (1/2), 41-47, 1998.

(XML 10) Extensible Markup Language (XML) <http://www.w3.org/XML/>



# APÉNDICE A. ESPECIFICACIÓN DE REQUISITOS SOFTWARE

Esta sección describe la especificación de requisitos software para el proyecto desarrollado. La especificación se ha hecho según en IEEE Std. 830-1998 (IEEE 98).

## 1 INTRODUCCIÓN

### 1.1 PROPÓSITO

El presente documento ha sido redactado con el fin de marcar las pautas generales y las especificaciones que deberá seguir el proyecto a desarrollar, con el objetivo final de suplir las necesidades planteadas para el mismo.

Este documento está dirigido a todas las personas con alguna relación en el proyecto con el objetivo de que realicen las funciones de desarrollo a partir de las pautas marcadas en él y en los documentos realizados posteriormente.

La Especificación de Requisitos servirá de canal de comunicación entre las distintas partes implicadas en el desarrollo de la aplicación y deberán tomar parte en su confección los miembros de cada una de estas partes. Esta especificación está sujeta a cambios que se recogerán por medio de sucesivas revisiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

### 1.2 ÁMBITO DE SISTEMA

Se ha constatado la necesidad de un sistema de Campus Virtual que sea independiente del LMSs utilizado (*Learning Management Systems*) a través de servicios web.

Así, dicho Campus Virtual se comunicará con los servicios Web. Cada servicio Web se conectará a una plataforma LMS con el objetivo de obtener toda la información que necesita el sistema del Campus virtual, así como de realizar las tareas de gestión sobre las plataformas de Campus Virtuales LMS.

En concreto, las plataformas LMS para las cuales se desarrollaran los servicios Web serán:

- Moodle
- Sakai

Las funcionalidades que debe soportar e implementar el Campus Virtual son las típicas de un sistema de Gestión de Campus virtuales: alta y baja de alumnos y profesores, alta y baja de asignaturas, foros, gestión de los contenidos, etc. Por ello, deberá distinguir entre los distintos tipos de usuarios (alumnos, profesores y administradores) e identificar el uso que puede hacer cada uno de ellos en el Campus Virtual.

### 1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

#### DEFINICIONES:

- **Campus Virtual:** Sistema utilizado para facilitar el proceso docente y la interacción entre profesores, investigadores y alumnos. ...

- **LMS:** Learning Management System. Herramientas de gestión de Campus Virtuales. Ejemplos: Blackboard, WebCT, Moodle, Sakai, Ilias, etc.
- **Plataforma:** LMS que da soporte a las funciones de un campus virtual
- **BD unidad apoyo campus virtual:** BBDD con los datos de los profesores, alumnos y asignaturas de una entidad docente que son susceptibles de participar del campus virtual de dicha entidad.
- **Foro:** Un foro está compuesto por temas, que a su vez se componen de mensajes.
- **Sesión:** Duración de una conexión entre un usuario y el sistema
- **Participante:** Usuario que ha sido dado de alta (en el ámbito de este proyecto, en asignaturas).
- **N/A:** No aplicable

#### ABREVIATURAS

- **SRS:** Especificación de Requisitos Software. Este documento.
- **CV:** Campus Virtual
- **LMS:** Learning Management System
- **BD:** Base de datos

#### 1.4 REFERENCIAS

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.

#### 1.5 VISIÓN GENERAL DEL DOCUMENTO

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.



## 2. DESCRIPCIÓN GENERAL

### 2.1. PERSPECTIVA DEL PRODUCTO

La perspectiva del proyecto es crear un Campus Virtual que interaccione directamente con una serie de Servicios Web que se comunicarán con las distintas plataformas LMS, en concreto, Moodle y Sakai.

### 2.2. FUNCIONES DEL PRODUCTO

El Campus Virtual soportará una conexión segura por http. Dependiendo del perfil de usuario con el que se acceda, se habilitarán unas u otras opciones en el Campus Virtual.

El Campus Virtual soportará la conexión simultánea de varios usuarios, permitiendo, en el caso de las opciones que lo posibiliten, la interacción entre los mismos.

Existirá una Web en la que, tras acceder con un determinado perfil, se podrán realizar las tareas básicas de gestión de un Campus Virtual.

Por otro lado, existirá un servicio web por cada plataforma LMS que se integre en el Campus Virtual. El cometido de este servicio web es el de interactuar entre la plataforma LMS y el Campus Virtual, eliminando la dependencia directa que había entre ellas.

La idea es tener un LMS que de soporte al campus virtual, pero ocultarlo totalmente al usuario.

Las funcionalidades que se podrán realizar serán básicamente:

- Alumno:
  - Acceso y salida.
  - Gestión de claves, que incluye la modificación de los datos de usuario.
  - Consulta de asignaturas.
  - Consulta de contenidos.
  - Participación en foros.
  - Consulta de anuncios
- Profesor: a las funcionalidades de un alumno, se añadirán:
  - Gestión de contenidos.
  - Gestión de anuncios.
  - Gestión de foros.
  - Alta asignatura.
  - Gestión de la asociación entre asignaturas y alumnos.
  - Gestión de la asociación entre asignaturas y profesores.
  - Consulta de usuarios de asignaturas.
  - Consulta de usuarios.
  - Consulta de asignaturas.
- Administrador:
  - Alta y baja de usuarios (alumnos y profesores)
  - Baja asignatura.

### 2.3. CARACTERÍSTICAS DE LOS USUARIOS

Existen tres tipos de usuarios:

- Alumno
- Profesor: Puede realizar un conjunto de tareas propias y todas las pertenecientes a un usuario de tipo Alumno.
- Administrador: Puede realizar un conjunto de tareas propias y todas las pertenecientes a un usuario de tipo Alumno.



## 2.4. RESTRICCIONES

**Lenguajes y Tecnologías en uso:** J2EE (Web Services), PHP, Apache, MySQL, XML, HTML.

Todas las conexiones al Campus Virtual se realizarán por conexión segura Http.

## 2.5. SUPOSICIONES Y DEPENDENCIAS

El sistema debe correr sobre cualquier plataforma gracias a la máquina virtual de java (JVM). Se realizará la página de registro y administración de usuarios cumpliendo los estándares para permitir la visualización en al menos IE, Firefox, Opera y Google Chrome.

## 2.6. REQUISITOS FUTUROS

- Ampliación de funcionalidades:
  - realización de exámenes online
  - incorporación de chat dentro de cada asignatura
  - realización de encuestas
  - fusión de asignaturas incluyendo contenidos, mensajes de foros, etc
- Integración con otras plataformas LMS
- Mejora de la Interfaz Web.
- Creación de más perfiles

### 3. REQUISITOS ESPECÍFICOS

#### 3.1. INTERFACES EXTERNAS

Debido a que la comunicación se realiza a través de servicios web, se utiliza SOAP, protocolo de comunicación bajo HTTP mediante mensajes XML. La estructura de mensajes de este protocolo establece 3 componentes diferenciados, el sobre, la cabecera y el cuerpo del mensaje. El cuerpo de cada mensaje contendrá contenido que se desea comunicar, debido a la dificultad para el envío de cadenas de objetos (arrays), el contenido del mensaje se establece como una cadena de caracteres para todas las funcionalidades que se permiten.

Otro requisito indispensable para el correcto funcionamiento y administración del Campus Virtual, además de la base de datos de la plataforma subyacente, es la existencia de una base de datos adicional. Esta base de datos adicional, designada como Base de Datos Campus Virtual (BBDD CV) contendrá los datos básicos e identificativos de los componentes necesarios para la correcta administración del campus virtual.

La base de datos Campus Virtual se representa como una mínima expresión de la GEA BD. Se compone de los datos identificativos para dos tipos de usuarios, alumnos y profesores, que se distinguen con claridad al almacenarlos en tablas distintas; asignaturas, actividades y grupos, a su vez forman una relación que indica la actividad de un grupo de una asignatura que ha sido virtualizada o tiene permisos para hacerlo por el profesor asignado, esta asignación se refleja en una relación entre profesor y grupo. Así mismo, los alumnos se relacionan con la actividad del grupo de cada asignatura en la que estén matriculados.

Debido al alcance de este proyecto, la base de datos de la plataforma específica no se utilizará en su totalidad. La estructura de la base de datos que se utiliza corresponde a la información completa de:

- Usuarios; alumnos y profesores.
- Asignaturas o cursos registrados.

Dentro de cada asignatura, se distinguen:

- Usuarios de cada asignatura.
- Estructura de contenidos de cada asignatura
- Contenidos (PDFs, DOCs, JPGs, etc.)
- Estructura de foros (temas) de cada asignatura
- Mensajes pertenecientes a cada tema
- Anuncios (mensajes) en cada asignatura

Entre ambas bases de datos existirá, en todo momento, total coherencia y consistencia, ya que las operaciones que se realicen tendrán repercusiones en ambas.

## 3.2. FUNCIONES

### GENERAL

#### 1. Acceso (Login).

Los usuarios del Campus Virtual deben introducir sus datos, los cuales serán validados para tener acceso al mismo.

Identificador	<b>01</b>
Función	<b>Acceso</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El sistema validará la cuenta del usuario, permitiendo el acceso a las diferentes funciones, restringidas por el tipo de usuario que realice esta operación.
Entrada	Identificador del usuario, Contraseña
Salida	Sesión creada
Origen	Interfaz
Destino	Interfaz
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Identificar al usuario</b>
Precondición	N/A
Postcondición	Usuario identificado de tipo Alumno (o superior)
Efectos colaterales	El sistema permite el acceso al resto de funciones que puede realizar el tipo de usuario identificado.

#### 2. Salida (Logout).

Los usuarios del Campus Virtual deben finalizar su sesión de trabajo al terminar de realizar sus operaciones.

Identificador	<b>02</b>
Función	<b>Salida</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El sistema cerrará la sesión del usuario, bloqueando el acceso al Campus Virtual
Entrada	Ninguna
Salida	Ninguna
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Cerrar la sesión del usuario</b>
Precondición	Usuario identificado de tipo Alumno (o superior)
Postcondición	El usuario no puede realizar ninguna operación, a excepción del acceso
Efectos colaterales	Sesión cerrada y eliminada

### 3. Gestión de claves: preguntas desafío.

Los usuarios del Campus Virtual podrán modificar la pregunta y respuesta desafío, que servirán para modificar la contraseña de acceso al campus virtual en caso de pérdida u olvido. Esta contraseña deberá ser actualizada tanto en la BD del Campus Virtual, como en la BD de la plataforma subyacente.

Identificador	<b>03</b>
Función	<b>ModificarDatos</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá modificar cualquiera o todos los datos (modificables) de su cuenta de usuario, que se componen de: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Email</li> <li>• Contraseña</li> <li>• Pregunta desafío y respuesta desafío con la cuales podrá recuperar su contraseña en caso de pérdida u olvido.</li> </ul>
Entrada	Nombre, Apellidos, Email, Contraseña, Pregunta desafío, Respuesta desafío.
Salida	Ninguna
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Modificar la pregunta y respuesta desafío</b>
Precondición	Usuario identificado de tipo Alumno (o superior)
Postcondición	Pregunta y respuesta desafío actualizadas
Efectos colaterales	Ninguno.

Identificador	<b>04</b>
Función	<b>ModificarContraseña</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá modificar la contraseña de acceso al Campus Virtual, después de haber respondido, a la pregunta desafío, con la respuesta desafío correcta, en caso de pérdida u olvido.
Entrada	Respuesta desafío, nueva contraseña
Salida	Ninguna
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Modificar la contraseña del usuario</b>
Precondición	Usuario identificado de tipo Alumno (o superior)
Postcondición	Contraseña modificada
Efectos colaterales	N/A

## ALUMNO/A

Además de los requisitos generales, los alumnos podrán:

### 1. Listado de asignaturas

Los alumnos podrán consultar sus asignaturas matriculadas, dadas de alta en la plataforma previamente.

Identificador	<b>05</b>
Función	<b>ListaAsignaturasUsuario</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá consultar el listado de asignaturas en las que se encuentra como participante (es decir, ha sido dado de alta previamente).
Entrada	Identificador de usuario
Salida	Listado de asignaturas del usuario
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar listado de asignaturas del usuario</b>
Precondición	Usuario identificado de tipo Alumno (o superior)
Postcondición	N/A
Efectos colaterales	N/A

### 2. Consulta de contenidos de cada asignatura

Cada asignatura contará con una serie de contenidos accesibles a los alumnos únicamente en modo lectura.

Identificador	<b>06</b>
Función	<b>ContenidosAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá obtener un listado de los recursos existentes en una asignatura elegida, de la que sea participante
Entrada	Identificador Asignatura
Salida	Listado de recursos existentes en la asignatura
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar listado de recursos de una asignatura</b>
Precondición	Usuario identificado de tipo Alumno (o superior)
Postcondición	N/A
Efectos colaterales	N/A

### 3. Participación en las asignaturas

#### a. Foros

Los alumnos podrán participar en los foros de cada asignatura que cuente con él, insertando nuevos temas o mensajes o leyendo los ya existentes. No se permitirá el borrado de ningún mensaje ya publicado.

Identificador	<b>07</b>
Función	<b>AgregarTema</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá agregar un tema al foro de una asignatura que imparta.
Entrada	Identificador asignatura, Nuevo tema
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Agregar un tema al foro de una asignatura</b>
Precondición	Usuario identificado de tipo Alumno (o superior) y participante de la asignatura
Postcondición	Nuevo tema creado
Efectos colaterales	Ampliado el listado de temas del foro de la asignatura

Identificador	<b>08</b>
Función	<b>AgregarMensaje</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá agregar nuevos mensajes a los temas del foro de una asignatura en la que participe.
Entrada	Asignatura, Tema, Título del mensaje, Mensaje
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Agregar nuevo mensaje a un tema del foro de una asignatura</b>
Precondición	Usuario identificado de tipo Alumno (o superior) y participante de la asignatura
Postcondición	Nuevo mensaje agregado al tema seleccionado de la asignatura
Efectos colaterales	Ampliado el listado de mensajes del tema del foro de la asignatura

Identificador	<b>09</b>
Función	<b>MensajesTemaForo</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener el listado de los mensajes pertenecientes a un tema seleccionado del foro de la asignatura elegida, en la que es participante.
Entrada	Asignatura, Tema

Salida	Listado de mensajes pertenecientes al tema elegido.
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar listado de mensajes de un tema del foro de una asignatura.</b>
Precondición	Usuario identificado de tipo Alumno (o superior) y participante de la asignatura.
Postcondición	N/A
Efectos colaterales	N/A

Identificador	<b>10</b>
Función	<b>TemasForoAsignatura</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener el listado de los temas pertenecientes al foro de la asignatura elegida, en la que es participante.
Entrada	Asignatura
Salida	Listado de temas pertenecientes al foro de la asignatura elegida
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar listado de temas del foro de una asignatura.</b>
Precondición	Usuario identificado de tipo Alumno (o superior) y participante de la asignatura.
Postcondición	N/A
Efectos colaterales	N/A

b. Anuncios

Los alumnos podrán visualizar los anuncios existentes en cada asignatura en la que participen.

Identificador	<b>11</b>
Función	<b>AvisosAsignatura</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener un listado de avisos pertenecientes a la asignatura elegida, en la que es participante.
Entrada	Identificador de asignatura
Salida	Listado de avisos pertenecientes a la asignatura elegida
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar listado de avisos de una asignatura.</b>
Precondición	Usuario identificado de tipo Alumno (o superior) y participante de la asignatura.
Postcondición	N/A
Efectos colaterales	N/A



## PROFESOR/A

Además de las funciones anteriores, los profesores podrán realizar:

### 1. Gestión de contenidos:

#### a. Administración de archivos

Los profesores podrán subir o borrar contenidos dentro de cada asignatura en las que sean profesores.

Identificador	<b>12</b>
Función	<b>AgregarRecurso</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá añadir un nuevo contenido a una asignatura que imparte.
Entrada	Identificador asignatura, Nuevo recurso
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Agregar un nuevo archivo a una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Nuevo recurso creado en la asignatura
Efectos colaterales	Ampliado el listado de recursos de la asignatura

Identificador	<b>13</b>
Función	<b>EliminarRecurso</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá eliminar recursos de una asignatura que imparte.
Entrada	Identificador asignatura, Identificador recurso
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Eliminar un archivo de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Recurso eliminado
Efectos colaterales	Reducido el listado de recursos de la asignatura

## b. Anuncios

Los profesores podrán publicar avisos dentro de una asignatura (en la que sean profesores) en forma de texto, que serán visibles para todos los alumnos de la asignatura, al igual que eliminarlos.

Identificador	<b>14</b>
Función	<b>AgregarAnuncio</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá añadir un nuevo aviso en una asignatura que imparte.
Entrada	Identificador asignatura, Nuevo anuncio
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Agregar un nuevo aviso en una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Nuevo aviso creado en la asignatura
Efectos colaterales	Ampliado el listado de avisos de la asignatura

Identificador	<b>15</b>
Función	<b>EliminarAnuncio</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá eliminar avisos en una asignatura que imparte.
Entrada	Asignatura, Aviso
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Eliminar un aviso en una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Aviso eliminado
Efectos colaterales	Reducido el listado de avisos de la asignatura

## c. Foros

Los profesores podrán borrar temas en un foro de una asignatura (en la que sean profesores), así como también borrar mensajes.

Identificador	<b>16</b>
Función	<b>EliminarMensaje</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá eliminar un mensaje de un tema del foro de una asignatura que imparta.
Entrada	Identificador asignatura, Tema, Mensaje
Salida	N/A
Origen	Interfaz

Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Eliminar un mensaje de un tema del foro de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Mensaje eliminado del tema seleccionado de la asignatura
Efectos colaterales	Reducido el listado de mensajes del tema del foro de la asignatura

Identificador	<b>17</b>
Función	<b>EliminarTema</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá eliminar un tema del foro de una asignatura que imparta.
Entrada	Identificador signatura, Tema
Salida	N/A
Origen	Interfaz
Destino	BD de la plataforma subyacente
Necesita	BD de la plataforma subyacente
Acción	<b>Eliminar un tema del foro de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) en la asignatura
Postcondición	Tema eliminado
Efectos colaterales	Reducido el listado de temas del foro de la asignatura

## 2. Alta asignatura en plataforma

Se podrán dar de alta nuevas asignaturas en la plataforma.

Identificador	<b>18</b>
Función	<b>NuevaAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá virtualizar una nueva asignatura que imparta, siempre que no esté previamente virtualizada por otro usuario; en dicho caso, el usuario deberá ser dado de alta en dicha asignatura.
Entrada	Identificador asignatura
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Virtualizar una nueva asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior)
Postcondición	Asignatura creada
Efectos colaterales	Aumentado el listado de asignaturas virtualizadas

### 3. Alta alumno en una asignatura

Los profesores podrán dar de alta nuevos alumnos en las asignaturas que impartan.

Identificador	<b>19</b>
Función	<b>AltaAlumnoAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá dar de alta a alumnos en una asignatura que imparta.
Entrada	Identificador asignatura, Identificador usuario
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Agregar alumno a una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) de la asignatura. Usuario, a agregar, existente. Asignatura existente.
Postcondición	Alumno asignado a una asignatura
Efectos colaterales	Aumentado el listado de usuarios de la asignatura seleccionada. Aumentado el listado de asignaturas del usuario.

### 4. Alta profesor en una asignatura

Los profesores podrán dar de alta a nuevos usuarios en las asignaturas que impartan, con el perfil de “profesor”.

Identificador	<b>20</b>
Función	<b>AltaProfesorAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá dar de alta a un profesor en una asignatura que imparta.
Entrada	Identificador asignatura, Identificador usuario
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Agregar profesor a una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) de la asignatura. Usuario, a agregar, existente. Asignatura existente.
Postcondición	Profesor asignado a una asignatura
Efectos colaterales	Aumentado el listado de usuarios de la asignatura seleccionada. Aumentado el listado de asignaturas del usuario.

## 5. Baja alumno en una asignatura

Los profesores podrán dar de baja a alumnos en las asignaturas que imparten.

Identificador	<b>21</b>
Función	<b>BajaAlumnoAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá dar de baja a un alumno en una asignatura que imparta.
Entrada	Identificador asignatura, Identificador usuario
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Eliminar alumno de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) de la asignatura. Alumno de la asignatura.
Postcondición	Alumno eliminado de la asignatura
Efectos colaterales	Reducido el listado de usuarios de la asignatura seleccionada. Reducido el listado de asignaturas del usuario.

## 6. Baja profesor en una asignatura

Los profesores podrán dar de baja a otros profesores en las asignaturas que imparten.

Identificador	<b>22</b>
Función	<b>BajaProfesorAsignatura</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá dar de baja a un profesor de una asignatura que imparta.
Entrada	Identificador asignatura, Identificador usuario
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Eliminar profesor de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) de la asignatura. Profesor de la asignatura.
Postcondición	Profesor eliminado de la asignatura
Efectos colaterales	Reducido el listado de usuarios de la asignatura seleccionada. Reducido el listado de asignaturas del usuario.

## 7. Listado de usuarios de una asignatura

Los profesores podrán consultar los usuarios de las asignaturas que impartan, dadas de alta en la plataforma previamente.

Identificador	<b>23</b>
Función	<b>UsuariosAsignatura</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener un listado de los usuarios participantes en una asignatura que imparta.
Entrada	Identificador asignatura
Salida	Listado de usuarios de una asignatura especificada.
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar el listado de usuarios de una asignatura</b>
Precondición	Usuario identificado de tipo Profesor (o superior) de la asignatura. Asignatura existente.
Postcondición	N/A
Efectos colaterales	N/A

## 8. Listado de usuarios

Los profesores podrán obtener un listado completo de todos los usuarios, dados de alta previamente en la plataforma.

Identificador	<b>24</b>
Función	<b>UsuariosCampusVirtual</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener un listado de los usuarios existentes en el Campus Virtual.
Entrada	N/A
Salida	Listado de usuarios existentes.
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar el listado de usuarios del Campus Virtual</b>
Precondición	Usuario identificado de tipo Profesor (o superior)
Postcondición	N/A
Efectos colaterales	N/A

## 9. Listado de asignaturas

Los profesores podrán consultar las asignaturas existentes, dadas de alta en la plataforma previamente.

Identificador	<b>25</b>
Función	<b>AsignaturasCampusVirtual</b>
Prioridad	Media
Estabilidad	Alta
Descripción	El usuario podrá obtener un listado de las asignaturas virtualizadas en el Campus Virtual.
Entrada	N/A
Salida	Listado de asignaturas virtualizadas.
Origen	Interfaz
Destino	Interfaz
Necesita	BD de la plataforma subyacente
Acción	<b>Consultar el listado de asignaturas virtualizadas del Campus Virtual</b>
Precondición	Usuario identificado de tipo Profesor (o superior)
Postcondición	N/A
Efectos colaterales	N/A

## ADMINISTRADOR DEL CAMPUS VIRTUAL

Además de las funciones anteriores, el administrador podrá realizar:

### 1. Alta alumno

El administrador podrá crear un nuevo usuario con permisos de alumno.

Identificador	<b>26</b>
Función	<b>AltaAlumno</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá crear una nueva cuenta de usuario con permisos de Alumno.
Entrada	Identificador usuario
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Agregar un nuevo usuario con permisos de Alumno.</b>
Precondición	Usuario identificado de tipo Administrador Identificador del usuario inexistente
Postcondición	Nueva cuenta de usuario de tipo Alumno creada.
Efectos colaterales	Aumentado el listado de usuarios existentes

## 2. Alta profesor

El administrador podrá crear un nuevo usuario con permisos de profesor.

Identificador	<b>27</b>
Función	<b>AltaProfesor</b>
Prioridad	Alta
Estabilidad	Alta
Descripción	El usuario podrá crear una nueva cuenta de usuario con permisos de Profesor.
Entrada	Identificador profesor
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Agregar un nuevo usuario con permisos de Profesor.</b>
Precondición	Usuario identificado de tipo Administrador Identificador del usuario inexistente
Postcondición	Nueva cuenta de usuario de tipo Profesor creada.
Efectos colaterales	Aumentado el listado de usuarios existentes

## 3. Baja alumno

El administrador podrá eliminar la cuenta de usuario de un alumno.

Identificador	<b>28</b>
Función	<b>BajaAlumno</b>
Prioridad	Baja
Estabilidad	Alta
Descripción	El usuario podrá eliminar los registros y la cuenta de un usuario de tipo Alumno
Entrada	Identificador alumno
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Eliminar un usuario con permisos de Alumno.</b>
Precondición	Usuario identificado de tipo Administrador Usuario de tipo Alumno existente
Postcondición	Usuario eliminado
Efectos colaterales	Reducido el listado de usuarios existentes



#### 4. Baja profesor

El administrador podrá eliminar la cuenta de usuario de un profesor.

Identificador	<b>29</b>
Función	<b>BajaProfesor</b>
Prioridad	Baja
Estabilidad	Alta
Descripción	El usuario podrá eliminar los registros y la cuenta de un usuario de tipo Profesor
Entrada	Identificador profesor
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Eliminar un usuario con permisos de Profesor.</b>
Precondición	Usuario identificado de tipo Administrador Usuario de tipo Profesor existente
Postcondición	Usuario eliminado
Efectos colaterales	Reducido el listado de usuarios existentes

#### 5. Eliminar asignatura

El administrador podrá eliminar una asignatura del Campus Virtual.

Identificador	<b>30</b>
Función	<b>EliminarAsignatura</b>
Prioridad	Baja
Estabilidad	Alta
Descripción	El usuario podrá eliminar una asignatura existente.
Entrada	Identificador asignatura
Salida	N/A
Origen	Interfaz
Destino	BD del Campus Virtual y BD de la plataforma subyacente
Necesita	BD del Campus Virtual y BD de la plataforma subyacente
Acción	<b>Eliminar una asignatura.</b>
Precondición	Usuario identificado de tipo Administrador Asignatura existente
Postcondición	Asignatura eliminada.
Efectos colaterales	Reducido el listado de asignaturas existentes.

### 3.3. REQUISITOS DE RENDIMIENTO

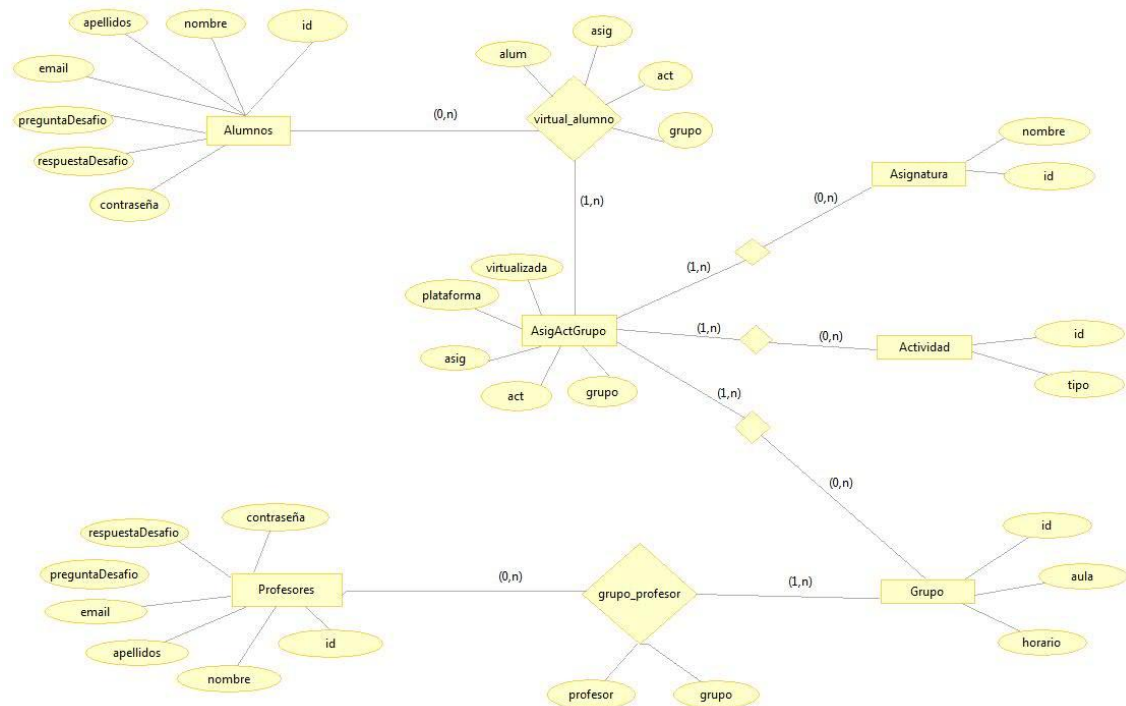
El servidor que soporta el sistema del campus virtual deberá ser capaz de dar servicio a múltiples alumnos y profesores simultáneamente, permitiendo la interacción entre ellos en los casos necesarios.

### 3.4. RESTRICCIONES DE BASE DE DATOS LÓGICA

Se requiere una base de datos relacional. Al ser una aplicación multiusuario esta deberá permitir el uso de transacciones, bloqueos, desbloques, permitir varias conexiones

simultáneamente, etc. Se utilizará como base de datos MySQL ya que las plataformas subyacentes utilizan este sistema y nos ofrece toda la funcionalidad necesaria para la implementación de una herramienta con acceso concurrente a los datos.

En la figura A.1 se muestra el diagrama de entidad-relación del esquema de la base de datos implementada.



**FIGURA A.1 DIAGRAMA E-R. BASE DE DATOS DEL CAMPUS VIRTUAL**

### 3.5. RESTRICCIONES DE DISEÑO

Se utilizarán patrones de desarrollo J2EE, y el lenguaje de implementación será Java.

### 3.6. ATRIBUTOS DEL SISTEMA

1. No podrán acceder al sistema usuarios no registrados.
2. Los alumnos no podrán realizar las tareas específicas de un profesor o de un administrador del Campus Virtual. Los profesores no podrán realizar las tareas específicas de un Administrador del Campus Virtual.
3. El sistema no funcionara en las plataformas que no dispongan de la máquina virtual de Java.



Los alumnos Ambrín Chaudhary Tabassam, Jaime Guimerá Coll y Gabriela Ruiz Escobar, como autores de este proyecto, autorizan a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria, documentación y el código generado a lo largo de este proyecto.

Ambrín Chaudhary  
Tabassam

Jaime Guimerá Coll

Gabriela Ruiz Escobar